Proceedings of the
10th Japanese-Hungarian Symposium
on Discrete Mathematics and Its Applications
May 22-25, 2017, Budapest, Hungary

Editors:

András Frank
Department of Operations Research
Eötvös Loránd University
frank@cs.elte.hu

András Recski
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
recski@cs.bme.hu

Gábor Wiener
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
wiener@cs.bme.hu

ISBN 978-963-313-253-1

Cover design: Kazuhiko Shiozaki, 1999

Contents

Preface

The present volume consists of the papers and extended abstracts of the talks presented at the 10th Japanese-Hungarian Symposium on Discrete Mathematics and its Applications (Budapest, May 22-25, 2017). Based on a long history of cooperation among Japanese and Hungarian scientists in the area of discrete mathematics, the previous symposia in this series took place in Kyoto (March 17-19, 1999), Budapest (April 20-23, 2001), Tokyo (January 21-24, 2003), Budapest (June 3-6, 2005), Sendai (April 3-5, 2007), Budapest (May 16-19, 2009), Kyoto (May 31 - June 3, 2011), Veszprém (June 4-7, 2013) and Fukuoka (June 2-5, 2015).
The 10th Symposium has been jointly organized by the Department of Operations Research, Eötvös Loránd University, Budapest and by the Department of Computer Science and Information Theory, Budapest University of Technology and Economics.

The conference has been supported by the Hungarian Academy of Sciences, by the National Research, Development and Innovation Office, by the Faculty of Science, Eötvös Loránd University and by the Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics.

The organizers wish to thank all the contributors for submitting papers, and all their colleagues, graduate students and sponsors for their assistance and support.

Recent progress on the network activation problem

Takuro Fukunaga

National Institute of Informatics
JST, ERATO,
Kawarabayashi Large Graph Project
takuro@nii.ac.jp

Abstract: In the network activation problem, each edge in a graph is associated with an activation function that decides whether the edge is activated from weights assigned to its end nodes. The feasible solutions of the problem are node weights, such that the activated edges form graphs of required connectivity, and the objective is to find a feasible solution minimizing its total weight. This problem includes the node-weighted network design problem, as well as several important applications motivated by communication networks. In this paper we introduce recent results on approximation algorithms for the network activation problem.

Keywords: network activation, survivable network design, spider covering algorithm

1 Introduction

The *network activation problem* is a problem of activating a well-connected network by assigning weights to nodes. The problem is formally described as follows. Given a graph $G = (V, E)$ and a set W of non-negative real numbers, a solution in the problem is a node weight function $w\colon V \to W$. For $u, v \in V$, let $\{u, v\}$ and uv denote the unordered and ordered pairs of u and v, respectively. Each edge $\{u, v\} \in E$ is associated with an activation function $\psi^{uv} : W \times W \to \{\text{true}, \text{false}\}$ such that $\psi^{uv}(i, j) = \psi^{vu}(j, i)$ holds for any $i, j \in W$. In this paper, each activation function ψ^{uv} is supposed to be *monotone*, i.e., if $\psi^{uv}(i, j) = \text{true}$ for some $i, j \in W$, then $\psi^{uv}(i', j') = \text{true}$ for any $i', j' \in W$ with $i' \geq i$ and $j' \geq j$. An edge $\{u, v\}$ is *activated* by w if $\psi^{uv}(w(u), w(v)) = \text{true}$. Let E_w be the set of edges activated by w in E. A node weight function w is feasible in the network activation problem if E_w satisfies given constraints, and the objective of the problem is to find a feasible node weight function w that minimizes $\sum_{v \in V} w(v)$, denoted by $w(V)$. We assume without loss of generality that $0 \in W$. We also assume throughout the paper that G is undirected even though the problem can be defined for directed graphs as well.

In this paper, we pose connectivity constraints on the set E_w of activated edges. Namely, we are given demand pairs $\{s_1, t_1\}, \ldots, \{s_d, t_d\} \subseteq V$ associated with connectivity requirements $r_1, \ldots, r_d$ defined as natural numbers. $[d]$ denotes $\{1, \ldots, d\}$, k denotes $\max_{i \in [d]} r_i$, and a node that participates in some demand pair is called a *terminal*. The constraints require that the connectivity between s_i and t_i in the graph (V, E_w) is at least r_i for each $i \in [d]$. We consider three definitions of connectivity: edge-connectivity, node-connectivity, and element-connectivity. The edge-connectivity between two nodes u and v is the maximum number of edge-disjoint paths between u and v, and the node-connectivity between u and v is the maximum number of inner disjoint paths between u and v. The element-connectivity is defined only for pairs of terminals, and for two terminals u and v, it is defined as the maximum number of paths between them that are disjoint in edges and in non-terminal nodes. The edge-connectivity network activation problem denotes the problem with the edge-connectivity constraints. The node- and the element-connectivity network activation problems are defined similarly.

The network activation problem is closely related to the *survivable network design problem* (SNDP), a problem of constructing a cheap network that is sufficiently connected. A feasible solution to the SNDP

is a subgraph (V, F) of a given graph $G = (V, E)$ that satisfies the connectivity constraints. There are two popular variations, called the edge- and node-weighted SNDPs. In the edge-weighted SNDP, each edge in the graph is associated with a weight $w(e)$, and the objective is to minimize the weight $w(F)$ of F defined as $\sum_{e \in F} w(e)$. In the node-weighted SNDP, a weight $w(v)$ is given for each node $v \in V$, and the objective is to minimize $\sum_{v \in V(F)} w(v)$, where $V(F)$ denotes the set of end nodes of edges in F. We denote $\sum_{v \in V(F)} w(v)$ by $w(V(F))$ in the sequel. It is known that the node-weighted SNDP generalizes the edge-weighted SNDP.

It can be seen that the network activation problem extends the node-weighted SNDP. Given node weights $w' \colon V \to \mathbb{R}_{\geq 0}$, let $W = \{w'(v) \colon v \in V\} \cup \{0\}$, and define a monotone activation function ψ^{uv} for $\{u, v\} \in E$ so that $\psi^{uv}(i, j) =$ true if and only if $i \geq w'(u)$ and $j \geq w'(v)$. A minimal solution $w \colon V \to W$ to the network activation problem with these activation functions does not assign a weight larger than $w'(v)$ to $v \in V$. Hence, if an edge activated by w is incident to a node v, then $w(v) = w'(v)$ holds without loss of generality. Therefore, the node-weighted SNDP with w' is equivalent to the network activation problem with ψ defined from w'.

The extension from the SNDP to the network activation problem is not only important from a technical viewpoint but also for practical reasons. In the node-weighted SNDP, for each node, one is required to decide whether it is chosen. In contrast, the network activation problem demands a decision concerning which weight is assigned to a node. In other words, the network activation problem admits more than two choices while the node-weighted SNDP admits only two choices for each node. This rich structure of the network activation problem enables to capture many problems motivated by realistic applications. In fact, Panigrahi [18] discussed numerous applications to wireless networks. In wireless networks, the success of communication between two base stations depends on factors such as physical obstacles between them, positions of antennas, and signal strength. Panigrahi suggested that many problems related to wireless networks can be modeled by the network activation problem. Moreover, the author and Maehara [8] observed that a problem of constructing a network with less monitoring cost of link failures is formulated as the network activation problem.

In this paper, we review recent results on a prize-collecting version of the network activation problem given in [7]. In the prize-collecting network activation problem (PCNAP), each demand pair $\{s_i, t_i\}$ is associated with not only a connectivity requirement r_i, but also a non-negative real number π_i, which is called the *penalty*. The edge set E_w activated by a solution w is allowed to violate the connectivity requirements, but it has to pay the penalty π_i if it does not satisfy the connectivity requirement for $\{s_i, t_i\}$. The objective of the problem is to minimize the sum of $w(V)$ and the penalties we have to pay. The author gave in [7] the first nontrivial algorithms for this problem. They relies on several new findings such as a nontrivial linear programming (LP) relaxation of the problem, a primal-dual algorithm for computing a subgraph called spider, and a potential function for analyzing a greedy algorithm. We briefly introduce these results in this paper.

The rest of this paper is organized as follows. In Section 2, we review related work on the network activation problem. In Section 3, we present a brief overview of the results obtained in [7]. In Section 4, we conclude the paper by mentioning several open problems.

2 Related work

The SNDP is a well-studied optimization problem, and there are substantial number of studies regarding algorithms for it. The best known approximation factors for the edge-weighted SNDP are 2 for the edge- [10] and element-connectivity [5], and $O(k^3 \log |V|)$ for node-connectivity [4]. For the node-weighted SNDP, Nutov [14] gave an $O(k \log |V|)$-approximation algorithm with edge-connectivity requirements, and element-connectivity requirements in [15]. His algorithm is based on an algorithm for the problem of covering uncrossable biset families by edges, where a biset is an ordered pair of two node sets, and an uncrossable family is a family closed under some uncrossing operations (we will present their formal definitions later). However, his analysis of the algorithm for covering uncrossable biset families has an error (see [7]).

The prize-collecting SNDP has also been well studied. As for edge-weighted graphs, we refer to only Hajiaghayi et al. [9] whereas many papers studied related problems such as the prize-collecting Steiner tree and forest. Recently much attention has been paid to node-weighted graphs. Könemann, Sadeghian, and Sanità [12] gave an $O(\log |V|)$-approximation algorithm for the prize-collecting node-weighted Steiner tree problem. Their algorithm has the Lagrangian multiplier preserving property, which is useful in many contexts. They also pointed out a technical error in Moss and Rabani [13]. Bateni, Hajiaghayi, and Liaghat [1] gave an $O(\log |V|)$-approximation algorithm for the prize-collecting node-weighted Steiner forest problem with application to the budgeted Steiner tree problem. Chekuri, Ene, and Vakilian [3] gave an $O(k^2 \log |V|)$-approximation for the prize-collecting SNDP with edge-connectivity requirements, which they later improved to $O(k \log |V|)$-approximation and also extended to the element-connectivity requirements (refer to [19]). We note that the proof in [19] implies that the algorithm in [15] works for the node-weighted SNDP with element-connectivity requirements, as Nutov originally claimed, even though his analysis of the algorithm for covering uncrossable biset families is not correct in general. We also note that the algorithm for the element-connectivity requirements in [19] implies $O(k^4 \log |V|)$-approximation for node-connectivity requirements, using the reduction from node-connectivity requirements to the element-connectivity requirements presented by Chuzhoy and Khanna [4].

Concerning the network activation problem, Panigrahi [18] gave $O(\log |V|)$-approximation algorithms for $k \leq 2$ and proved that it is NP-hard to obtain an $o(\log |V|)$-approximation algorithm even when activated edges are required to be a spanning tree. Nutov [17] presented approximation algorithms for higher connectivity requirements, including $O(k \log |V|)$-approximation for the edge- and element-connectivity and $O(k^4 \log^2 |V|)$-approximation for the node-connectivity. He also discussed special node-connectivity requirements such as rooted and subset requirements. These results are built based on his research in [15] for covering uncrossable biset families. This contains an error as mentioned above, and the rectification offered in [19] cannot be extended to the network activation problem. Therefore, the network activation problem had no non-trivial algorithms for the element- and node-connectivity before the author's work [7].

An important factor in most of the research mentioned above is the *greedy spider cover algorithm*. The notion of *spiders* was invented by Klein and Ravi [11] in order to solve the node-weighted Steiner tree problem. It was originally defined as a tree that admits at most one node of degree larger than two and that spans at least two terminals. The node of degree larger than two is called the *head*, and nodes of degree one are called the *feet* of the spider. It is supposed without loss of generality that each foot of a spider is a terminal. If all nodes have degrees of at most two, then an arbitrary node is chosen to be the head. Klein and Ravi [11] proved that any Steiner tree can be decomposed into node-disjoint spiders so that each terminal is included in some spider. The *density* of a subgraph is defined as its node weight divided by the number of terminals included in it. The decomposition theorem implies that there exists a spider with a density of at most that of Steiner trees. Since contracting a spider with f feet decreases the number of terminals by at least $f - 1$, a greedy algorithm to repeatedly contract minimum density spiders achieves $O(\log |V|)$-approximation. Minimum density spiders are hard to compute but their relaxations can be computed by a simple algorithm that involves first guessing the place of the head and number of feet, which is possible because there are only $|V|$ options for each. Let h be the head, and f be the number of feet. We then compute a shortest path from h to each terminal, and choose the f shortest paths from them. The union of these shortest paths is not necessarily a spider, but its density is at most that of spiders, and contracting the union can play the same role as contracting spiders. Nutov [14, 15, 17] extended the notion of spiders to uncrossable biset families, and demonstrated in the sequence of his research that they are useful for the node-weighted SNDP and the network activation problem.

3 Prize-collecting network activation problem

In this section, we give an overview of approximation algorithms for PCNAP given in [7]. Algorithms given in [7] achieve $O(k \log |V|)$-approximation for the edge-connectivity PCNAP, and $O(k^2 \log |V|)$-

Table 1: Approximation factors for the edge-weighted SNDP, node-weighted SNDP, and the network activation problem

	non-prize-collecting		prize-collecting	
edge-connectivity				
edge-weighted SNDP	2	Jain [10]	2.54	Hajiaghayi et al. [9]
node-weighted SNDP	$O(k \log \lvert V \rvert)$	Nutov [14]	$O(k \log \lvert V \rvert)$	Chekuri et al. [3]
network activation	$O(k \log \lvert V \rvert)$	Nutov [17]	$O(k \log \lvert V \rvert)$	Fukunaga [7]
element-connectivity				
edge-weighted SNDP	2	Fleischer et al. [5]	2.54	Hajiaghayi et al. [9]
node-weighted SNDP	$O(k \log \lvert V \rvert)$	Vakilian [19]	$O(k \log \lvert V \rvert)$	Vakilian [19]
network activation	$O(k^2 \log \lvert V \rvert)$	Fukunaga [7]	$O(k^2 \log \lvert V \rvert)$	Fukunaga [7]

approximation for the element-connectivity PCNAP. Table 1 summarizes the approximation factors achieved by these algorithms and other related studies. Using decompositions of connectivity requirements given in [4], we can also achieve $O(k^5 \log^2 |V|)$-approximation for the node-connectivity PCNAP. These results give the first non-trivial algorithms for the PCNAP. We also recall that, besides these algorithms, no algorithms were known even for the element- and node-connectivity network activation problems. For wireless networks, it is natural to consider node-connectivity, which represents tolerance against node failures, rather than edge-connectivity, which represents tolerance against link failures. Hence, these results are important for not only theory but also applications.

Let us present a high level overview of these algorithms. The algorithms first reduce the problem with high connectivity requirements to the *augmentation problem*, which asks to increase the connectivity of demand pairs by one. This is a standard trick for SNDP, and the author showed that this trick can work even for the PCNAP. Then, the algorithms compute an optimal solution to an LP relaxation, and discards some of the demand pairs according to the optimal solution, which is a popular way to deal with prize-collecting problems since Bienstock et al. [2]. In the last step, the algorithms solves the problem using the greedy spider cover algorithm. To obtain an approximation guarantee, it is required to show that the minimum density of spiders can be bounded in terms of the optimal value of the LP relaxation. This is achieved by presenting a primal-dual algorithm for computing spiders, which is the same approach as [3, 1, 19].

As observed from this overview, the algorithms rely on many ideas given in the previous studies on the prize-collecting SNDP and the network activation problem. However, it is highly nontrivial to apply these ideas for the PCNAP, and it requires several new ideas to obtain the algorithms. Specifically, the technical contributions of [7] are based on the following three new findings: an LP relaxation of the problem, a primal-dual algorithm for computing spiders, and a potential function for analyzing the greedy spider cover algorithm. Below we explain these one by one.

LP relaxation

Nutov's spider decomposition theorem is useful for the biset covering problem defined from the SNDP and the network activation problem, but we have to strengthen it for solving their prize-collecting versions. We define an LP relaxation of the problem and compare the minimum density of spiders with the density of fractional solutions feasible to this relaxation. The same attempt has been made previously by [1, 3, 12] for the node-weighted SNDP, but our situation is much more complicated. Each connectivity requirement in the node-weighted SNDP can be simply represented by demands on the number of chosen nodes in node cuts of graphs, which naturally formulates an LP relaxation that performs well. On the other hand, the network activation problem requires the decision of which edges are activated for covering bisets in addition to the decision on which weights are assigned to nodes for activating the edges. Hence an LP relaxation for the network activation problem needs variables corresponding to edges and nodes whereas that for the node-weighted SNDP needs only variables corresponding to nodes. However, dealing with

both edge and node variables introduces a large integrality gap into a natural LP relaxation for the network. Hence we require to formulate an LP relaxation carefully.

In [7], the author proposed a new LP that lifts the natural LP relaxation for the PCNAP. It is non-trivial even to see that the LP relaxes the PCNAP. The author proved it using the structure of biset families defined from the connectivity constraints, wherein the biset family can be decomposed into a polynomial number of ring biset families, and the degree of each node is at most two in any minimal edge cover of a ring biset family.

The idea on formulating the LP relaxation is potentially useful for other covering problems. The author pointed out in [6] that a natural LP relaxation has a large integrality gap for many covering problems in node-weighted graphs. He also presented several tight approximation algorithms using the LP relaxations designed based on the idea proposed in [7].

Primal-dual algorithm for computing spiders

For bounding the minimum density of spiders in terms of optimal values of our relaxation, the author presented a primal-dual algorithm for computing spiders. Usually, a primal-dual algorithm computes fractional solutions feasible to the dual of an LP relaxation together with primal solutions, but this seems difficult for the relaxation because of its complicated form. Hence, the algorithm does not directly compute solutions feasible to the dual of our relaxation. Instead, another LP simpler than our relaxation is defined, and the algorithm computes feasible solutions to the dual of this simpler LP. Although the simpler LP does not relax our relaxation, we can show that it is within a constant factor of the relaxation if biset families are restricted to laminar families of cores, which are bisets that do not include more than one minimal biset. The primal-dual algorithm computes dual solutions that assign non-zero values only to variables corresponding to cores in laminar families. Hence, the density of spiders can be analyzed in terms of our relaxation.

Summarizing, the algorithm uses two different LPs: the LP obtained by lifting the natural relaxation is used for deciding which demand pairs are discarded in the first step, and the simpler LP with laminar core families is used in the second step that iterates choosing spiders. We note that the simpler LP cannot be used in the first step because of two reasons. First, we do not know beforehand which laminar core families will be used, and second, we have different laminar families in distinct iterations.

Although the primal-dual algorithm for the simpler LP seems to be similar to primal-dual algorithms known for related problems, its design and analysis is not trivial. One reason for this is the existence of more than one choice of weights for each end node of activated edges as we have already mentioned. Another reason is the involved structure of bisets. Since a biset is defined as an ordered pair of two node sets, covering a biset family by edges is a much more difficult problem than covering a set family, for which primal-dual algorithms are often studied. Indeed, the algorithm utilizes many non-trivial properties of uncrossable biset families. Vakilian [19] also studied a primal-dual algorithm for computing a spider on an uncrossable biset family, but his algorithm uses a property of biset families arising from node-weighted SNDP with element-connectivity requirements. On the other hand, the algorithm of [7] deals with arbitrary uncrossable biset families.

Potential function for analyzing greedy spider cover algorithm

Nutov [15] claimed that repeatedly choosing a constant approximation of minimum density spiders achieves $O(\log |V|)$-approximation for covering uncrossable biset families. This claim is true if biset families are defined from edge-connectivity requirements. However it is not true for all uncrossable biset families. The claim is based on the fact that contracting a spider with f feet decreases the number of minimal bisets by a constant fraction of f. However there is a case in which contracting a spider does not decrease the number at all. Chekuri, Ene, and Vakilian [19] showed that the claim is true for biset families arising from the node-weighted SNDP, but it cannot be extended to arbitrary uncrossable biset families, including those from the network activation problem.

To rectify this situation, a new potential function was introduced in [7]. This potential function depends on the number of minimal bisets and nodes shared by at least two minimal bisets. If the number

of minimal bisets does not decrease considerably when a spider is selected, many new minimal bisets share the head of the spider. This fact motivates the definition of the potential function.

With this new potential function, the definition of density of an edge set will be changed to the total weight for activating it divided by the value of the potential function. We cannot prove that the minimum density of spiders is at most that of biset family covers after changing the definition of density. Instead, we will show that a spider minimizing the density in the old definition approximates the density of biset family covers in the new definition within a factor of $O(k)$. This proves that the greedy spider covering algorithm achieves $O(k \log |V|)$-approximation for the biset covering problem with uncrossable biset families. Since Klein and Ravi [11], the greedy spider cover algorithms have been applied to many problems related to the node-weighted SNDP. Considering this usefulness of the greedy spider cover algorithms, the potential function is of independent interest because it is required for analyzing the algorithms for uncrossable biset families.

4 Conclusion

In this paper, we reviewed the results on approximation algorithms for PCNAP given in [7]. The algorithms are built on new formulations of LP relaxations, the primal-dual algorithm for computing spiders, and the potential function for analyzing the greedy spider cover algorithm.

There are several important open problems on network activation problem, and let us mention a few of them. One open problem is to improve the approximation factor for the element-connectivity network activation problem. The author gave an $O(k^2 \log |V|)$-approximation algorithm for this problem, but this approximation factor is worse than that known for the node-weighted SNDP by a factor k. It should be interesting if the approximation factor can be improved to $O(k \log |V|)$.

Another open problem is the existence of efficient algorithms for the network activation problem on restricted graph classes. In particular, algorithms for the Steiner tree activation problem on unit disk graphs are important because the node-weighted Steiner tree problem is studied actively for the unit disk graphs in a context of wireless network operation. As for the unit disk graphs, it is open whether or not there exists a constant-factor approximation algorithm even for the Steiner Steiner tree activation problem, a special case of the Steiner activation problem in which the connectivity requirements demand that all given terminals are connected by activated edges while the node-weighted Steiner tree admits a constant-factor on unit disk graphs. In [8], the author and Maehara gave a constant-factor approximation algorithm for a vertex-cover-weighted Steiner tree problem, which is a special case of the Steiner tree activation problem, on unit disk graphs. However, this algorithm is already complicated, and its approximation factor is a very large constant. Hence we believe that a novel idea is required for obtaining an efficient approximation algorithm for the Steiner tree activation problem on unit disk graphs.

References

[1] M. Bateni, M. Hajiaghayi, and V. Liaghat. Improved approximation algorithms for (budgeted) node-weighted Steiner problems. In *ICALP (1)*, vol. 7965 of *Lecture Notes in Computer Science*, pages 81–92, 2013.

[2] D. Bienstock, M. X. Goemans, D. Simchi-Levi, and D. P. Williamson. A note on the prize collecting traveling salesman problem. *Mathematical Programming*, 59:413–420, 1993.

[3] C. Chekuri, A. Ene, and A. Vakilian. Prize-collecting survivable network design in node-weighted graphs. In *APPROX-RANDOM*, vol. 7408 of *Lecture Notes in Computer Science*, pages 98–109, 2012.

[4] J. Chuzhoy and S. Khanna. An $O(k^3 \log n)$-approximation algorithm for vertex-connectivity survivable network design. *Theory of Computing*, 8(1):401–413, 2012.

[5] L. Fleischer, K. Jain, and D. P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *Journal of Computer and System Sciences*, 72(5):838–867, 2006.

[6] T. Fukunaga. Covering problems in edge- and node-weighted graphs. *Discrete Optimization*, 20:40–61, 2016.

[7] T. Fukunaga. Spider covers for prize-collecting network activation problem. In *SODA*, pages 9–24, 2015.

[8] T. Fukunaga, T. Maehara. Computing a tree having a small vertex cover. In *COCOA*, vol. 10043 of *Lecture Notes in Computer Science*, pages 77–91, 2016.

[9] M. T. Hajiaghayi, R. Khandekar, G. Kortsarz, and Z. Nutov. Prize-collecting steiner network problems. *ACM Transactions on Algorithms*, 9(1):2, 2012.

[10] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[11] P. N. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted Steiner trees. *Journal of Algorithms*, 19(1):104–115, 1995.

[12] J. Könemann, S. S. Sadeghabad, and L. Sanità. An LMP $O(\log n)$-approximation algorithm for node weighted prize collecting Steiner tree. In *FOCS*, pages 568–577, 2013.

[13] A. Moss and Y. Rabani. Approximation algorithms for constrained node weighted Steiner tree problems. *SIAM Journal on Computing*, 37(2):460–481, 2007.

[14] Z. Nutov. Approximating Steiner networks with node-weights. *SIAM Journal on Computing*, 39(7):3001–3022, 2010.

[15] Z. Nutov. Approximating minimum-cost connectivity problems via uncrossable bifamilies. *ACM Transactions on Algorithms*, 9(1):1, 2012.

[16] Z. Nutov. Approximating subset k-connectivity problems. *Journal of Discrete Algorithms*, 17:51–59, 2012.

[17] Z. Nutov. Survivable network activation problems. *Theoretical Computer Science*, 514:105–115, 2013.

[18] D. Panigrahi. Survivable network design problems in wireless networks. In *SODA*, pages 1014–1027, 2011.

[19] A. Vakilian. Node-weighted prize-collecting survivable network design problems. Master's thesis, University of Illinois at Urbana-Champaign, 2013.

The maximum vanishing subspace problem, CAT(0)-space relaxation, and block-triangularization of partitioned matrices (extended abstract)

Hiroshi Hirai[1]

Department of Mathematical Informatics,
Graduate School of Information Science and Technology,
The University of Tokyo,
Tokyo, 113-8656, Japan.
hirai@mist.i.u-tokyo.ac.jp

Abstract: In this paper we address the following algebraic generalization of the bipartite stable set problem. We are given a block matrix $A = (A_{\alpha\beta})$, where $A_{\alpha\beta}$ is an m_α by n_β matrix over field $\mathbf{F}$ for $\alpha = 1, 2, \ldots, \mu$ and $\beta = 1, 2, \ldots, \nu$. The maximum vanishing subspace problem (MVSP) is to find vector subspaces $X_\alpha \subseteq \mathbf{F}^{m_\alpha}$ and $Y_\beta \subseteq \mathbf{F}^{n_\beta}$ such that each $A_{\alpha\beta} : \mathbf{F}^{m_\alpha} \times \mathbf{F}^{n_\beta} \to \mathbf{F}$ vanishes on $X_\alpha \times Y_\beta$, and the sum $\sum_\alpha \dim X_\alpha + \sum_\beta \dim Y_\beta$ of their dimension is maximum. This problem arises from a study of a canonical block-triangular form of A by Ito, Iwata, and Murota (1994).

We prove that MVSP can be solved in polynomial time. Our proof is a novel combination of submodular optimization on modular lattices and convex optimization on CAT(0)-spaces. We present implications of this result for block-triangulations of A.

This is a joint work with Masaki Hamada.

Keywords: CAT(0)-space, proximal point algorithm, Dulmage-Mendelsohn decomposition, partitioned matrix, submodular function, modular lattice.

1 Introduction

The maximum stable set problem in bipartite graphs is one of the fundamental and well-solved combinatorial optimization problems. In this paper we address the following algebraic generalization of the bipartite stable set problem. We are given a matrix A partitioned into submatrices as

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1\nu} \\ A_{21} & A_{22} & \cdots & A_{2\nu} \\ \vdots & \vdots & \ddots & \vdots \\ A_{\mu 1} & A_{\mu 2} & \cdots & A_{\mu\nu} \end{pmatrix},$$

where $A_{\alpha\beta}$ is an $m_\alpha \times n_\beta$ matrix over field $\mathbf{F}$ for $\alpha = 1, 2, \ldots, \mu$, $\beta = 1, 2, \ldots, \nu$. Such a matrix is called a *partitioned matrix of type* $(m_1, m_2, \ldots, m_\mu; n_1, n_2, \ldots, n_\nu)$. The *maximum vanishing subspace problem (MVSP)* is to maximize

$$\sum_{\alpha=1}^{\mu} \dim X_\alpha + \sum_{\beta=1}^{\nu} \dim Y_\beta \tag{1.1}$$

[1]Research is supported by JSPS KAKENHI Grant Numbers 25280004, 26330023, 26280004,17K00029.

over vector subspaces $X_\alpha \subseteq \mathbf{F}^{m_\alpha}$ $(\alpha = 1, 2, \ldots, m)$, $Y_\beta \subseteq \mathbf{F}^{n_\beta}$ $(\beta = 1, 2, \ldots, n)$ satisfying

$$A_{\alpha\beta}(X_\alpha, Y_\beta) = \{0\} \quad (1 \leq \alpha \leq \mu, 1 \leq \beta \leq \nu), \tag{1.2}$$

where each submatrix $A_{\alpha\beta}$ is regarded as a bilinear form $\mathbf{F}^{m_\alpha} \times \mathbf{F}^{n_\beta} \to \mathbf{F}$ by

$$(u, v) \mapsto u^\top A_{\alpha\beta} v. \tag{1.3}$$

A subspace $(X_1, X_2, \ldots, X_\mu, Y_1, Y_2, \ldots, Y_\nu)$ is called *vanishing* if it satisfies (1.2), and is called *maximum* if it attains the maximum of (1.1).

MVSP generalizes the maximum stable set problem on bipartite graphs. Indeed, consider the case $m_\alpha = n_\beta = 1$ for each α, β. Namely each submatrix is a scalar. Then each vector subspace is $\{0\}$ or $\mathbf{F}$, and its dimension is 0 or 1. The condition (1.2) says that one of X_α and Y_β is $\{0\}$ if $A_{\alpha\beta}$ is a nonzero scalar. Consider a bipartite graph on vertices $a_1, a_2, \ldots, a_\mu, b_1, b_2, \ldots b_\nu$ such that edge $a_\alpha b_\beta$ is given if and only if $A_{\alpha\beta}$ is a nonzero scalar. Then MVSP is nothing but the maximum stable set problem on this bipartite graph.

A linear algebraic interpretation of MVSP is explained as follows. Consider a transformation of A with form

$$\begin{pmatrix} E_1^\top & O & \cdots & O \\ O & E_2^\top & \ddots & \vdots \\ \vdots & \ddots & \ddots & O \\ O & \cdots & O & E_\mu^\top \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1\nu} \\ A_{21} & A_{22} & \cdots & A_{2\nu} \\ \vdots & \vdots & \ddots & \vdots \\ A_{\mu 1} & A_{\mu 2} & \cdots & A_{\mu\nu} \end{pmatrix} \begin{pmatrix} F_1 & O & \cdots & O \\ O & F_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & O \\ O & \cdots & O & F_\nu \end{pmatrix}, \tag{1.4}$$

where E_α is a nonsingular $m_\alpha \times m_\alpha$ matrix for $\alpha = 1, 2, \ldots, \mu$ and F_β is a nonsingular $n_\beta \times n_\beta$ matrix for $\beta = 1, 2, \ldots, \nu$. If the resulting matrix contains a zero submatrix of c rows and d columns, then from the corresponding rows and columns, we obtain a vanishing subspace of dimension $c + d$. Conversely, from a vanishing subspace of dimension b, we can find a transformation of form (1.4) such that the resulting matrix contains a zero submatrix of c rows and d columns with $c + d = b$. Thus MVSP is the problem of finding a transformation (1.4) of A such that the resulting matrix has a zero submatrix of largest size.

Ito, Iwata, and Murota [13] studied a canonical block triangular-form under transformation (1.4), which generalizes the classical *Dulmage-Mendelsohn decomposition* [7, 8]; see also [17]. They formulated an equivalent problem of MVSP, though MVSP was formally introduced by a recent paper [11]. For several basic special cases [7, 8, 11, 19], MVSP can be solved in polynomial time via Gaussian elimination, bipartite matching, and matroid intersection algorithm, and a canonical block-triangular form is also obtained accordingly. These works are in a cross road of numerical computation and combinatorial optimization. Ito, Iwata, and Murota [13, p.1252] raised an open problem of solving (an equivalent problem of) MVSP in polynomial time. The main result of this paper solves this open problem.

Theorem 1.1. *MVSP can be solved in polynomial time.*

Significances, implications, and a novel proof technique of this result are explained as follows.

Submodular optimization on modular lattice. MVSP is viewed a submodular function minimization (SFM) on the lattice of all vector subspaces of a vector space. Such a lattice is a typical instance of a *modular lattice*. Submodular optimization on modular lattice is a new emerging field in combinatorial optimization. Kuivinen [15] proved a good characterization of SFM on the product $\mathcal{L}^n$ of a modular lattice $\mathcal{L}$, where $\mathcal{L}$ is finite, and is a part of an input. In this setting, Fujishige, Király, Makino, Takazawa, and Tanigawa [9] proved the oracle-tractability when $\mathcal{L}$ is a modular lattice of rank 2. In the valued-CSP setting where a submodular function is given as a sum of submodular functions with few number variables, a tractability criterion of Kolmogorov, Thapper, and Živný [14] implies that SFM on $\mathcal{L}^n$ is solved in polynomial time. In contrast with these results, our SFM is on an *infinite* modular lattice ruled out by a linear algebraic machinery. To the best of our knowledge, Theorem 1.1 is the first positive result on such a discrete optimization problem over an infinite lattice of vector subspaces.

Beyond Euclidean convexity: outline of the proof. No reasonable LP/convex relaxation (allowing infiniteness) is known for MSVP. This is a reason of the difficulty. Beyond Euclidean convexity, our proof method employs a method of a *non-Euclidean convex optimization*, more specifically, *convex optimization on CAT(0)-space.* Here a *CAT(0)-space* is a nonpositively-curved metric space enjoying various fascinating properties analogous to those in Euclidean space; see [5]. One of important features of a CAT(0)-space is the unique geodesic property: every pair of points can be joined by the unique geodesic. Through the unique geodesics, several convexity concepts (e.g., convex functions) are naturally introduced. Computational and algorithmic theory on CAT(0)-space is also an emerging research field; see e.g., [1, 2, 21]. Our proof method connects the convexity of CAT(0)-spaces with the polynomial time complexity in discrete optimization.

As is well-known, a (usual) submodular function on Boolean lattice $\{0,1\}^n$ is extended to a convex function on hypercube $[0,1]^n$ in Euclidean space, via *Lovász extension* [16]. This fact enables us to apply a Euclidean convex optimization method (e.g., the ellipsoid method) to various problems related to the submodular function. Analogous to $\{0,1\}^n \hookrightarrow [0,1]^n$, a modular lattice $\mathcal{L}$ is embedded into a suitable continuous metric space $K(\mathcal{L})$, called the *orthoscheme complex* [4]. It is shown in [6, 10] that $K(\mathcal{L})$ is a CAT(0)-space. In this setting, a submodular function is extended to a convex function on $K(\mathcal{L})$ [12]. Consequently, our problem MVSP becomes a convex optimization over a CAT(0)-space.

We will solve this continuous optimization problem by utilizing a CAT(0)-space version of a *proximal point algorithm (PPA)*. The Euclidean PPA is a well-known simple iterative algorithm to minimize a convex function f, which computes the proximal point operator $J^f_\lambda(z)$ of the current point z, updates $z \leftarrow J^f_\lambda(z)$, and repeat. The PPA is naturally defined on a CAT(0)-space. Bačák [2] showed that the sequence (z_ℓ) generated by PPA converges to a minimizer of f; see also [3]. We apply a version of PPA to our CAT(0)-space relaxation of MVSP. By using a recent result of Ohta and Palfia [20] on the rate of the convergence, we show that after a polynomial number of iterations, a maximum vanishing space is obtained from the current point z_ℓ. We finally show that the proximal operator in each step is computed in polynomial time. This is the most technical but intriguing part of the proof.

Block-triangulation of partitioned matrix. Let us return the original motivation of MVSP. A maximal chain of maximum vanishing subspaces provides, via the change of base, the most refined block-triangulation under transformation (1.4), which we call the *DM-decomposition* [11, 13]. Solving MVSP is not enough to obtaining the DM-decomposition. We here introduce a reasonably *coarse* block-triangulation, which we call a *quasi DM-decomposition*. A quasi DM-decomposition still generalizes known important special cases, such as CCF [19]. We show that a quasi DM-decomposition can be obtained in polynomial time by solving a weighted version of MVSP with varying weights. We think that obtaining a quasi DM-decomposition is a limit which we can do. The difference between DM-decomposition and quasi DM-decomposition seems to be a matter of numerical analysis/computation; obtaining DM-decomposition solves the common invariant subspace problem, which is an extremely difficult problem in numerical computation.

References

[1] F. Ardila, M. Owen, and S. Sullivant, Geodesics in CAT(0) cubical complexes, *Advances in Applied Mathematics* **48** (2012), 142–163

[2] M. Bačák, The proximal point algorithm in metric spaces, *Israel Journal of Mathematics* **194** (2013), 689–701.

[3] M. Bačák, *Convex Analysis and Optimization in Hadamard Spaces*. De Gruyter, Berlin, 2014.

[4] T. Brady and J. McCammond, Braids, posets and orthoschemes. *Algebraic and Geometric Topology* **10** (2010), 2277–2314.

[5] M. R. Bridson and A. Haefliger, *Metric Spaces of Non-positive Curvature*. Springer-Verlag, Berlin, 1999.

[6] J. Chalopin, V. Chepoi, H. Hirai, and D. Osajda. Weakly modular graphs and nonpositive curvature. (2014), arXiv:1302.5877.

[7] A. L. Dulmage and N. S. Mendelsohn, Coverings of bipartite graphs. *Canadian Journal of Mathematics* **10** (1958), 517–534.

[8] A. L. Dulmage and N. S. Mendelsohn, A structure theory of bipartite graphs of finite exterior dimension, *Transactions of the Royal Society of Canada, Section III* **53** (1959), 1–13.

[9] S. Fujishige, T Király, K. Makino, K. Takazawa, and S. Tanigawa, Minimizing Submodular Functions on Diamonds via Generalized Fractional Matroid Matchings. EGRES Technical Report (TR-2014-14), (2014).

[10] T. Haettel, D. Kielak, and P. Schwer, The 6-strand braid group is CAT(0). *Geometriae Dedicata*, to appear.

[11] H. Hirai, Computing DM-decomposition of a partitioned matrix with rank-1 blocks. (2016), arXiv:1609.01934.

[12] H. Hirai, L-convexity on graph structures. (2016), arXiv:1610.02469.

[13] H. Ito, S. Iwata, and K. Murota, Block-triangularizations of partitioned matrices under similarity/equivalence transformations. *SIAM Journal on Matrix Analysis and Applications* **15** (1994), 1226–1255.

[14] V. Kolmogorov, J. Thapper, and S. Živný, The power of linear programming for general-valued CSPs. *SIAM Journal on Computing*, **44** (2015), 1–36.

[15] F. Kuivinen, On the complexity of submodular function minimisation on diamonds. *Discrete Optimization*, **8** (2011), 459–477.

[16] L. Lovász, Submodular functions and convexity. In A. Bachem, M. Grötschel, and B. Korte (eds.): *Mathematical Programming—The State of the Art* (Springer-Verlag, Berlin, 1983), 235–257.

[17] L. Lovász and M. Plummer, *Matching Theory*, North-Holland, Amsterdam, 1986.

[18] K. Murota, *Matrices and Matroids for Systems Analysis.* Springer-Verlag, Berlin, 2000.

[19] K. Murota, M. Iri, and M. Nakamura, Combinatorial canonical form of layered mixed matrices and its application to block-triangularization of systems of linear/nonlinear equations. *SIAM Journal on Algebraic and Discrete Methods* **8** (1987), 123–149.

[20] S. Ohta and M. Pálfia, Discrete-time gradient flows and law of large numbers in Alexandrov spaces. *Calculus of Variations and Partial Differential Equations* **54** (2015) 1591–1610.

[21] M. Owen, Computing geodesic distances in tree space, *SIAM Journal on Discrete Mathematics* **25** (2011), 1506–1529.

The Weighted Linear Matroid Parity Problem[1]

Satoru Iwata[2]

Department of Mathematical Informatics
University of Tokyo
Tokyo 113-8656, Japan
iwata@mist.i.u-tokyo.ac.jp

Yusuke Kobayashi[3]

Division of Policy and Planning Sciences
University of Tsukuba
Tsukuba, Ibaraki, 305-8573, Japan
kobayashi@sk.tsukuba.ac.jp

Abstract: The matroid parity (or matroid matching) problem, introduced as a common generalization of matching and matroid intersection problems, is so general that it requires an exponential number of oracle calls. Lovász (1980) showed that this problem admits a min-max formula and a polynomial algorithm for linearly represented matroids. Since then efficient algorithms have been developed for the linear matroid parity problem.

We present a combinatorial, deterministic, polynomial-time algorithm for the weighted linear matroid parity problem. The algorithm builds on a polynomial matrix formulation using Pfaffian and adopts a primal-dual approach based on the augmenting path algorithm of Gabow and Stallmann (1986) for the unweighted problem.

Keywords: Linear matroid parity, matching, polynomial-time algorithm, Pfaffian, primal-dual approach

1 Introduction

The matroid parity problem [12] (also known as the matchoid problem [11] or the matroid matching problem [13]) was introduced as a common generalization of matching and matroid intersection problems. In the worst case, it requires an exponential number of independence oracle calls [10, 15]. Nevertheless, Lovász [13, 15, 16] showed that the problem admits a min-max theorem for linear matroids and presented a polynomial algorithm that is applicable if the matroid in question is represented by a matrix.

Since then, efficient combinatorial algorithms have been developed for this linear matroid parity problem [3, 18, 19]. Gabow and Stallmann [3] developed an augmenting path algorithm with the aid of a linear algebraic trick, which was later extended to the linear delta-matroid parity problem [5]. Orlin and Vande Vate [19] provided an algorithm that solves this problem by repeatedly solving matroid intersection problems coming from the min-max theorem. Later, Orlin [18] improved the running time bound of this algorithm. The current best deterministic running time bound due to [3, 18] is $O(nm^{\omega})$, where n is the cardinality of the ground set, m is the rank of the linear matroid, and ω is the matrix multiplication exponent, which is at most 2.38. These combinatorial algorithms, however, tend to be complicated.

An alternative approach that leads to simpler randomized algorithms is based on an algebraic method. This is originated by Lovász [14], who formulated the linear matroid parity problem as rank computation of a skew-symmetric matrix that contains independent parameters. Substituting randomly generated numbers to these parameters enables us to compute the optimal value with high probability. A straightforward adaptation of this approach requires iterations to find an optimal solution. Cheung, Lau, and

[1]The first author presented a prototype of our algorithm without a full proof in the 8th JHSDM [8]. The full version of our paper is now available in [9].

[2]Supported by JST, CREST and by KAKENHI No. 24106005 from MEXT.

[3]Supported by JST, ERATO, Kawarabayashi Large Graph Project, and by KAKENHI No. 24106002 from MEXT and No. 16K16010 from JSPS.

Leung [2] have improved this algorithm to run in $O(nm^{\omega-1})$ time, extending the techniques of Harvey [7] developed for matching and matroid intersection.

While matching and matroid intersection algorithms have been successfully extended to their weighted version, no polynomial algorithms have been known for the weighted linear matroid parity problem for more than three decades. Camerini, Galbiati, and Maffioli [1] developed a random pseudopolynomial algorithm for the weighted linear matroid parity problem by introducing a polynomial matrix formulation that extends the matrix formulation of Lovász [14]. This algorithm was later improved by Cheung, Lau, and Leung [2]. The resulting complexity, however, remained pseudopolynomial. Tong, Lawler, and Vazirani [21] observed that the weighted matroid parity problem on gammoids can be solved in polynomial time by reduction to the weighted matching problem.

We present a combinatorial, deterministic, polynomial-time algorithm for the weighted linear matroid parity problem. Note that a prototype of our algorithm was presented in [8]. In the algorithm, we combine algebraic approach and augmenting path technique together with the use of node potentials. The algorithm builds on a polynomial matrix formulation, which naturally extends the one discussed in [4] for the unweighted problem. The algorithm employs a modification of the augmenting path search procedure for the unweighted problem by Gabow and Stallmann [3]. It adopts a primal-dual approach without writing an explicit LP description. The correctness proof for the optimality is based on the idea of combinatorial relaxation for polynomial matrices due to Murota [17]. The algorithm is shown to require $O(n^3m)$ arithmetic operations. This leads to a strongly polynomial algorithm for linear matroids represented over a finite field. For linear matroids represented over the rational field, one can exploit our algorithm to solve the problem in polynomial time.

Independently of the present work, Gyula Pap has obtained another combinatorial, deterministic, polynomial-time algorithm for the weighted linear matroid parity problem based on a different approach (see [20]).

2 Our Result

Let A be a matrix of row-full rank over an arbitrary field $\mathbf{K}$ with row set U and column set V. Assume that both $m = |U|$ and $n = |V|$ are even. The column set V is partitioned into pairs, called *lines*. Each $v \in V$ has its *mate* $\bar{v}$ such that $\{v, \bar{v}\}$ is a line. We denote by L the set of lines, and suppose that each line $\ell \in L$ has a weight $w_\ell \in \mathbb{R}$.

The linear dependence of the column vectors naturally defines a matroid $\mathbf{M}(A)$ on V. Let $\mathcal{B}$ denote its base family. A base $B \in \mathcal{B}$ is called a *parity base* if it consists of lines. As a weighted version of the linear matroid parity problem, we will consider the problem of finding a parity base of minimum weight, where the weight of a parity base is the sum of the weights of lines in it. This problem generalizes finding a minimum-weight perfect matching in graphs and a minimum-weight common base of a pair of linear matroids on the same ground set.

As another weighted version of the matroid parity problem, one can think of finding a matching (independent parity set) of maximum weight. This problem can be easily reduced to the minimum-weight parity base problem.

Our main result is stated as follows.

Theorem 1 *There exists an algorithm that finds a parity base of minimum weight or detects infeasibility with $O(n^3m)$ arithmetic operations over $\mathbf{K}$.*

If $\mathbf{K}$ is a finite field of fixed order, each arithmetic operation can be executed in $O(1)$ time. Hence Theorem 1 implies the following.

Corollary 2 *The minimum-weight parity base problem over an arbitrary fixed finite field $\mathbf{K}$ can be solved in strongly polynomial time.*

When $\mathbf{K} = \mathbb{Q}$, it is not obvious that a direct application of our algorithm runs in polynomial time. This is because we do not know how to bound the number of bits required to represent the entries of

matrices appeared in the algorithm. However, the minimum-weight parity base problem over $\mathbb{Q}$ can be solved in polynomial time by applying our algorithm over a sequence of finite fields.

Theorem 3 *The minimum-weight parity base problem over $\mathbb{Q}$ can be solved in time polynomial in the binary encoding length $\langle A \rangle$ of the matrix representation A.*

We refer to the full paper [9] for the proofs.

References

[1] P. M. Camerini, G. Galbiati, and F. Maffioli: Random pseudo-polynomial algorithms for exact matroid problems, *J. Algorithms*, 13 (1992), 258–273.

[2] H. Y. Cheung, L. C. Lau, and K. M. Leung: Algebraic algorithms for linear matroid parity problems, *ACM Trans. Algorithms*, 10 (2014), 10: 1–26.

[3] H. N. Gabow and M. Stallmann: An augmenting path algorithm for linear matroid parity, *Combinatorica*, 6 (1986), 123–150.

[4] J. F. Geelen and S. Iwata: Matroid matching via mixed skew-symmetric matrices, *Combinatorica*, 25 (2005), 187–215.

[5] J. F. Geelen, S. Iwata, and K. Murota: The linear delta-matroid parity problem, *J. Combinatorial Theory*, Ser. B, 88 (2003), 377–398.

[6] D. Gijswijt and G. Pap: An algorithm for weighted fractional matroid matching, *J. Combinatorial Theory*, Ser. B, 103 (2013), 509–520.

[7] N. J. A. Harvey: Algebraic algorithms for matching and matroid problems, *SIAM J. Comput.*, 39 (2009), 679–702.

[8] S. Iwata: A weighted linear matroid parity algorithm, *Proceedings of the 8th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications*, pp. 251–259, 2013.

[9] S. Iwata and Y. Kobayashi: A weighted linear matroid parity algorithm, *Mathematical Engineering Technical Reports*, METR 2017-01, University of Tokyo, 2017.

[10] P. M. Jensen and B. Korte: Complexity of matroid property algorithms, *SIAM J. Comput.*, 11 (1982), 184–190.

[11] T. A. Jenkyns: *Matchoids: A Generalization of Matchings and Matroids*, Ph. D. Thesis, University of Waterloo, 1974.

[12] E. Lawler: *Combinatorial Optimization — Networks and Matroids*, Holt, Rinehalt, and Winston, 1976.

[13] L. Lovász: The matroid matching problem, *Algebraic Methods in Graph Theory*, Colloq. Math. Soc. János Bolyai, 25 (1978), 495–517.

[14] L. Lovász: On determinants, matchings, and random algorithms, *Fundamentals of Computation Theory*, L. Budach ed., Academie-Verlag, 1979, 565–574.

[15] L. Lovász: Matroid matching and some applications, *J. Combinatorial Theory*, Ser. B, 28 (1980), 208–236.

[16] L. Lovász: Selecting independent lines from a family of lines in a space, *Acta Sci. Math.*, 42 (1980), 121–131.

[17] K. Murota: Computing the degree of determinants via combinatorial relaxation, *SIAM J. Comput.*, 24 (1995), 765–796.

[18] J. B. Orlin: A fast, simpler algorithm for the matroid parity problem, *Proceedings of the 13th International Conference on Integer Programming and Combinatorial Optimization*, LNCS 5035, Springer-Verlag, 2008, 240–258.

[19] J. B. Orlin and J. H. Vande Vate: Solving the linear matroid parity problem as a sequence of matroid intersection problems, *Math. Programming*, 47 (1990), 81–106.

[20] G. Pap: Weighted linear matroid matching, *Proceedings of the 8th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications*, pp. 411–413, 2013.

[21] P. Tong, E. L. Lawler, and V. V. Vazirani: Solving the weighted parity problem for gammoids by reduction to graphic matching, *Progress in Combinatorial Optimization*, W. R. Pulleyblank, ed., Academic Press, 1984, 363–374.

Global Rigidity of Triangulations with Braces

Tibor Jordán[1]

Department of Operations Research
Eötvös University
MTA-ELTE Egerváry Research Group
Pázmány Péter sétány 1/C, 1117
Budapest, Hungary
jordan@cs.elte.hu

Shin-ichi Tanigawa[2]

Department of Mathematical Informatics
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, 113-8656,
Tokyo Japan
tanigawa@mist.i.u-tokyo.ac.jp

Abstract: The rigidity of polyhedra in 3-space is one of the central subjects in rigidity theory, whose history dates back to Cauchy's work. Cauchy's theorem implies that a convex simplicial polyhedron is rigid as a bar-and-joint framework. A natural question would be whether simplicial polyhedra have a stronger rigidity property such as global rigidity (i.e., unique realizability). Although Cauchy's theorem states uniqueness within the family of convex realizations, such a global rigidity property fails if we drop the assumption of convexity. In fact Hendrickson proved that the 1-skeleton of a generic simplicial polyhedron cannot be globally rigid, and thus it has to be braced by extra edges for the unique realizability.

In this paper we prove a simple combinatorial characterization of the global rigidity of the 1-skeleta of generic simplicial polyhedra with braces. We also discuss how it can be used to refine known rigidity properties of simplicial polyhedra.

Keywords: global rigidity, polyhedron, Cauchy's rigidity theorem

1 Introduction

The celebrated theorem of Cauchy states that if the vertex-edge graphs of two convex polyhedra are isomorphic and corresponding faces are congruent then the two polyhedra are the same. This theorem in particular implies that a convex simplicial polyhedron (i.e., a convex polyhedron with triangular faces) is rigid as a bar-and-joint framework. A natural question would be whether simplicial polyhedra have a stronger rigidity property, such as global rigidity (i.e., unique realizability). Cauchy's theorem states uniqueness within the family of convex realizations, but the uniqueness fails if we drop the assumption of convexity. For example if the graph of a simplicial polyhedron has a separator of size three, then one can always construct a distinct realization by reflecting one side of the polyhedron along the hyperplane spanned by those three points. Thus 4-connectivity is necessary for the global rigidity of 1-skeleta.

In 1992 B. Hendrickson [13] proved a necessary condition for a generic realization of a graph to be globally rigid, which in turn implies that the 1-skeleton of a generic polyhedron cannot be globally rigid regardless of the connectivity of the underlying graph. Motivated by this fact in this paper we consider simplicial polyhedra *braced* by extra edges. See Figure 1.

W. Whiteley proved that a simplicial polyhedron with a bracing edge has a substantially stronger rigidity property if the underlying graph is 4-connected.

Theorem 1 (Whiteley [19]) *A generic simplicial polyhedron with one bracing edge is redundantly rigid (i.e., rigid after the removal of any edge) in $\mathbb{R}^3$ if the underlying graph is 4-connected.*

[1]Research is supported by the National Research, Development and Innovation Office, grant no. NKFIH K115483 and K 109240.

[2]Research is supported by JSPS Postdoctoral Fellowships for Research Abroad, JSPS Grant-in-Aid for Scientific Research(A)(25240004), and JSPS Grant-in-Aid for Scientific Research (C) 15KT0109.

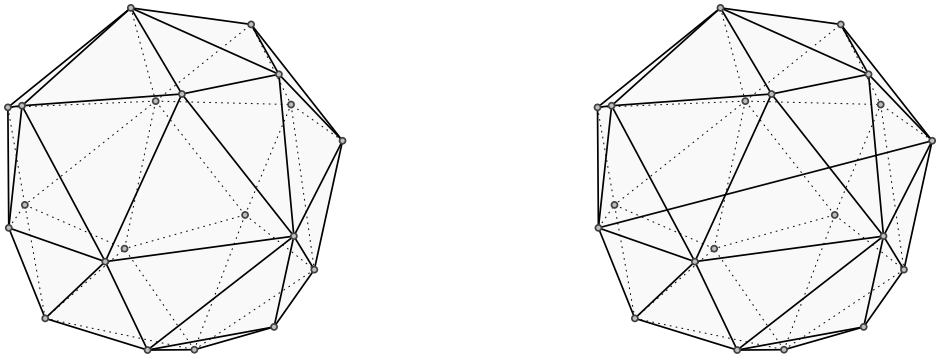


Figure 1: A simplicial polyhedron and a simplicial polyhedron with a brace.

In his talk at the Advances in Combinatorial and Geometric Rigidity Workshop (BIRS, Banff, 2015) he conjectured that every 4-connected uni-braced generic simplicial polyhedron is in fact globally rigid. In this work we prove the following more general statement.

Theorem 2 *A generic simplicial polyhedron with at least one bracing edge is globally rigid in* $\mathbb{R}^3$ *if the underlying graph is 4-connected.*

As we remarked above 4-connectivity is a trivial necessary condition for the global rigidity, and hence Theorem 2 chracterizes the global rigidity of generic simplicial polyhedra wth braces.

2 Further Results

2.1 Preliminaries

We first review basic terminologies from rigidity theory.

A d-dimensional *bar-and-joint framework* (or *framework*, for short) is a pair (G, p), where $G = (V, E)$ is a simple graph and p is a map from V to $\mathbb{R}^d$. We may think of the vertices as universal joints and the edges as rigid (i.e. fixed length) bars connecting certain pairs of joints. A framework (G, p) is said to be a *realization* of G in $\mathbb{R}^d$. We say that (G, p) is *rigid* in $\mathbb{R}^d$ if every continuous motion of its vertices in $\mathbb{R}^d$ which preserves all edge lengths takes the framework to a realization of G which is congruent to (G, p).

The 1-skeleton of a polyhedron P, with the given spatial positions of the vertices, gives rise to a three-dimensional framework, which is a realization of the graph $G(P)$ of a polyhedron. If P is a convex polyhedron with only triangular faces then this framework is rigid by Cauchy's rigidity theorem.

In this paper we are interested in global rigidity, a stronger property than rigidity. We say that two realizations (G, p) and (G, q) of a graph G are *equivalent* if $||p(u) - p(v)|| = ||q(u) - q(v)||$ holds for all pairs u, v with $uv \in E$, and *congruent* if $||p(u) - p(v)|| = ||q(u) - q(v)||$ holds for all pairs u, v with $u, v \in V$. Here $||.||$ denotes the Euclidean norm in $\mathbb{R}^d$. A d-dimensional framework (G, p) is *globally rigid* in $\mathbb{R}^d$ if every framework in $\mathbb{R}^d$ which is equivalent to (G, p) is congruent to (G, p). In other words, the edge lengths uniquely determine all pairwise distances. We say that (G, p) is *generic* if the set of the $d|V|$ coordinates of the vertices is algebraically independent over the rationals.

It is known that the rigidity (resp. the global rigidity) of frameworks in $\mathbb{R}^d$ is a generic property for every fixed dimension $d \geq 1$, that is, the rigidity (resp. global rigidity) of (G, p) depends only on the graph G and not the particular realization p, if (G, p) is generic, see [1, 12]. Thus we say that the graph G is *rigid* (resp. *globally rigid*) in $\mathbb{R}^d$ if every (or equivalently, if some) generic realization of G in $\mathbb{R}^d$ is rigid (resp. globally rigid).

It is well-known [11] that the graphs of the triangulated convex polyhedra are rigid in $\mathbb{R}^3$. This implies by a theorem of Steinitz that a *maximal planar graph* or a *planar triangulation* (or a *triangulations*, for

short) is rigid in $\mathbb{R}^3$. On the other hand, the following theorem by B. Hendrickson implies that a triangulation cannot be globally rigid in $\mathbb{R}^3$ as it cannot be redundantly rigid.

Theorem 3 (Hendrickson [13]) *Let G be globally rigid in $\mathbb{R}^d$. Then either G is a complete graph on at most $d+1$ vertices, or G is (d+1)-connected and redundantly rigid in $\mathbb{R}^d$, i.e., $G-e$ is rigid for every edge e in G.*

Indeed, the equivalence between rigidity and infinitesimal rigidity for generic bar-joint frameworks implies that $|E(G)| > 3|V(G)|-6$ is necessary for a graph G to be redundantly rigid in three-space [1], and hence a triangulation has to be "braced" by an extra edge to satisfy Hendrickson's necessary condition. In what follows we shall call a graph $H = (V, E + B)$ a *braced triangulation* if it is obtained from a triangulation $G = (V, E)$ by adding a set B of new edges (called *bracing edges*). In the special case when $|B| = 1$ we say that H is a *uni-braced* triangulation.

2.2 Refined rigidity properties of (braced) triangulations

By using the terminologies defined in the last subsection, Theorem 2 can be restated as follows.

Theorem 4 *Every 4-connected braced triangulation is globally rigid in $\mathbb{R}^3$.*

It follows from Theorem 4 that if a graph contains a triangulation as a spanning subgraph then Hendrickson's condition is necessary and sufficient to imply global rigidity in $\mathbb{R}^3$.

A pair of vertices $\{u, v\}$ in a framework (G, p) is *globally linked* in (G, p) if, in all equivalent frameworks (G, q), we have $||p(u) - p(v)|| = ||q(u) - q(v)||$. The pair $\{u, v\}$ is *globally linked* in G if it is globally linked in all generic frameworks (G, p). Thus G is globally rigid if and only if all pairs of vertices of G are globally linked. We say that a pair of vertices $\{u, v\}$ is *globally loose* in a graph G if $\{u, v\}$ is not globally linked in all generic realizations of G.

The following theorem is a stronger version of Hendrickson's theorem for simplicial polyhedra.

Theorem 5 *Let G be a triangulation and let $\{u, v\}$ be a pair of non-adjacent vertices of G. Then $\{u, v\}$ is globally loose.*

When a braced triangulation is not 4-connected, a natural question would be to identify or enumerate globally rigid subframeworks. Namely we are interesting in characterizing a *globally rigid cluster* of G, i.e, a maximal set of vertices of G in which each pair is globally linked. For this we need the following easy observation.

Lemma 6 *Let $G = (V, E)$ be a triangulation, let $H = (V, E + B)$ be a braced triangulation with $|B| \geq 1$, and let $ab \in B$. Let Y denote the set of vertices of H which can be separated from $\{a, b\}$ in H by a three-separator and let $X = V - Y$. Then $H[X]$ is the unique maximal 4-connected subgraph of H which contains ab.*

The subgraph $H[X]$ in the lemma is called the *4-block of ab in H*.

Theorem 7 *Let H be a braced triangulation. Then the globally rigid clusters of H are the vertex sets of the 4-blocks of the bracing edges as well as the maximal complete subgraphs of H not contained by any of these 4-blocks.*

It follows that every globally rigid cluster induces a globally rigid subgraph in a braced triangulation. Thus every braced triangulation has a globally rigid subgraph on at least five vertices. In a uni-braced triangulation it coincides with the fundamental circuit of the bracing edge with respect to G in the three-dimensional rigidity matroid.

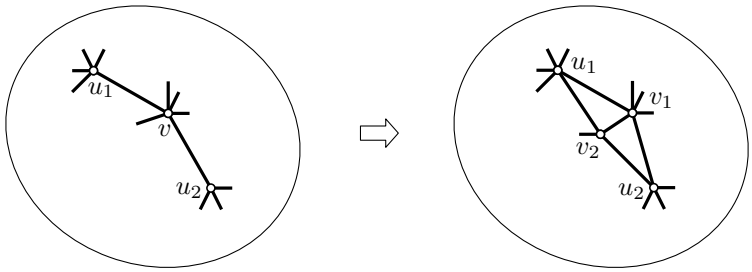


Figure 2: A 2-vertex splitting operation at v with $U_{01} = \{u_1, u_2\}$.

2.3 Redundant rigidity of braced triangulations

Since globally rigid graphs are redundantly rigid by Theorem 3, it follows from Theorem 4 that every 4-connected braced triangulation $G = (V, E)$ is redundantly rigid in $\mathbb{R}^3$. As we noted earlier, the special case of this corollary concerning uni-braced triangulations was proved by Whiteley [19, Theorem 5.3], using different methods.

We can refine these results and characterize redundantly rigid braced triangulations and redundant edges in braced triangulations as follows.

Theorem 8 *Let $H = (V, E + B)$ be a braced triangulation and let $e \in E + B$ be a designated edge. Then $H - e$ is rigid if and only if e belongs to the 4-block of some bracing edge in H.*

It follows that H is redundantly rigid if and only if every edge belongs to the 4-block of some bracing edge. This result is an extension of Whiteley's theorem [19] on block and hole strctures. See [10] for other kinds of extensions of Whiteley's theorem.

We close this subsection with some remarks on higher degrees of redundancy. Whiteley conjectured that if G is a 5-connected braced triangulation with $|E| = 3|V| - 4$ then removing any two bars from G leaves a (minimally) rigid graph [19, Conjecture 5.1]. Motivated by our new results we may strengthen this conjecture as follows.

Conjecture 9 *Let $G = (V, E)$ be a 5-connected braced triangulation with $|E| \geq 3|V| - 4$. Then $G - e$ is globally rigid in $\mathbb{R}^3$ for all $e \in E$.*

2.4 Vertex splitting

Theorem 4 will follow from an inductive construction of 4-connected uni-braced triangulations and a set of new results on the effect of the vertex splitting operation on the stresses of a generic framework. This operation is defined as follows.

Let $H = (V, E)$ be a graph. For a vertex $v \in V$ we use $N_H(v)$ to denote the set of neighbours of v in H. Given a vertex $v_1 \in V$ and a partition $\{U_{01}, U_0, U_1\}$ of $N_H(v)$ with $|U_{01}| = k$, the *k-vertex splitting* operation at v_1 with respect to $\{U_{01}, U_0, U_1\}$ removes the edges connecting v_1 to U_0 and inserts a new vertex v_0 as well as new edges between v_0 and $\{v_1\} \cup U_{01} \cup U_0$. See Figure 2.

The operation is *nontrivial* if U_0 and U_1 are both non-emtpy.

The vertex-splitting operation is well-known in rigidity theory as well as in the theory of polyhedra and triangulations of surfaces. Steinitz proved that every triangulation can be obtained from K_4 by a sequence of 2-vertex splitting operations. Whiteley [20] proved that $(d-1)$-vertex splitting preserves rigidity in $\mathbb{R}^d$.

Whiteley conjectured that $(d-1)$-vertex splitting preserves global rigidity in $\mathbb{R}^d$ provided it does not create vertices of degree d.

Conjecture 10 (Connelly and Whiteley [9]) *Let H be globally rigid in $\mathbb{R}^d$ with at least $d+2$ vertices and let G be obtained from H by a nontrivial $(d-1)$-vertex-splitting operation. Then G is globally rigid in $\mathbb{R}^d$.*

This conjecture is still open for $d \geq 3$ (a proof for $d = 2$ is given in [18]). Our second main result is the following.

Theorem 11 *Suppose that G can be obtained from K_{d+2} by a sequence of non-trivial $(d-1)$-vertex splitting operations. Then G is globally rigid in $\mathbb{R}^d$.*

Based on the new vertex-splitting result, we were also able to prove the global rigidity of triangulations on other surfaces.

Theorem 12 *Suppose that G is a 4-connected triangulation of the torus or the projective plane. Then G is globally rigid in $\mathbb{R}^3$.*

We remark that there exist infinitely many 4-connected triangulations of the torus containing no spanning triangulations of the plane (and hence they are not braced triangulations in the planar sense).

A natural open problem is whether global rigidity holds for triangulations on any 2-surface except for sphere.

3 Proof of Theorem 4

Theorem 4 follows rather easily once we can prove the statement for uni-braced triangulations. Theorem 4 for the uni-braced case follows from Theorem 11 and the following.

Theorem 13 *Let G be a 4-connected uni-braced triangulation. Then G can be obtained from K_5 by a sequence of non-trivial vertex splitting operations.*

In this abstract we only give a sketch of the proof of Theorem 11. Although the proof of Theorem 11 is based on the standard machinery from the theory of equilibrium stresses, we shall introduce a new notation, called the *degeneracy* of stresses, which may be of independent interest.

3.1 Equilibrium stresses

An *equilibrium stress* (or *stress*, for short) for a framework (G, p) in $\mathbb{R}^d$ is an assignment $\omega : E \to \mathbb{R}$ such that, for each vertex $v_i \in V$ we have

$$\sum_{j:v_iv_j \in E} \omega_{i,j}(p(v_i) - p(v_j)) = 0 \tag{1}$$

where we use $\omega_{i,j}$ for $\omega(v_i, v_j)$ for simplicity. The *stress matrix* Ω associated to ω is the $|V| \times |V|$ symmetric matrix in which the entries are defined so that $\Omega[i, j] = -\omega_{i,j}$ for all edges $v_iv_j \in E$, $\Omega[i, j] = 0$ for all non-adjacent vertex pairs $v_i, v_j \in V$, and $\Omega[i, i]$ is chosen so that each row and column sum is equal to zero. It is easy to verify that the rank of Ω is at most $|V| - d - 1$. We say that a stress matrix Ω is of *full rank* if its rank is equal to $|V| - d - 1$.

For generic frameworks, results of B. Connelly (sufficiency) and Gortler, Healy and Thurston (necessity) give rise to a characterization of global rigidity in terms of stress matrices.

Theorem 14 (Gortler, Healy and Thurston [12]) *Let (G, p) be a generic framework in $\mathbb{R}^d$ on at least $d + 2$ vertices. Then (G, p) is globally rigid in $\mathbb{R}^d$ if and only if (G, p) has an equilibrium stress ω for which the rank of the associated stress matrix Ω is $|V| - d - 1$.*

A corollary of Theorem 14 is the following sufficient condition for the global rigidity of a graph, due to Connelly and Whiteley.

Theorem 15 (Connelly and Whiteley [9]) *Suppose that a framework (G, p) with $|V(G)| \geq d + 2$ is infinitesimally rigid in $\mathbb{R}^d$ and admits a full rank stress matrix. Then G is globally rigid in $\mathbb{R}^d$.*

In view of Theorem 15, in order to prove the global rigidity of a graph G we may focus on finding a realization (G, p) that is infinitesimally rigid and admits a full rank stress matrix.

3.2 Nondegenerate Stresses

One of the key tools in our study of vertex splitting is the new notion of nondegenerate stress. It is defined as follows. Let (G,p) be a d-dimensional framework and let ω be a stress on (G,p). For a given vertex v of G and a given non-empty subset $X \subseteq N_G(v)$ we define $\omega \circ p(X) \in \mathbb{R}^d$ by

$$\omega \circ p(X) := \sum_{u \in X} \omega(uv)(p(u) - p(v)).$$

We say that ω is *degenerate* (resp. *nondegenerate*) with respect to a d-subpartition[1] $\{X_1, \ldots, X_d\}$ of $N_G(v)$ if the set of vectors $\{\omega \circ p(X_i) : 1 \leq i \leq d\}$ is linearly dependent (linearly independent, respectively). Due to the equilibrium condition, ω is always degenerate with respect to a d-partition of $N_G(v)$. We say that ω is *nondegenerate* if it is nondegenerate with respect to every vertex v and every proper d-subpartition of the neighborhood of v. In this subsection we prove some fundamental facts related to this notion.

We call a graph G *nondegenerate* in $\mathbb{R}^d$ if every generic realization (G,p) of G in $\mathbb{R}^d$ admits a nondegenerate stress. One can prove that nondegeneracy is a generic property in $\mathbb{R}^d$ for all $d \geq 1$.

A stress ω of a framework (G,p) is called *nowhere zero* if $\omega(e) \neq 0$ for every $e \in E(G)$. Note that if G is nondegenerate then every generic realization of G must admit a nowhere zero stress. The following stronger observation easily follows from the equilibrium condition.

Lemma 16 *Let G be a connected graph. If G is nondegenerate then it is M-connected.*

There exist M-connected graphs which are degenerate as it is shown by the following three-dimensional example. (The example was inspired by an observation by Connelly [8].) The graph G in Figure 3 is obtained by glueing two copies of K_5 along a triangle $\{1,2,3\}$ and deleting the edge 23. It is easy to check that G is an M-circuit in $\mathbb{R}^3$ and hence every generic realization of G has a unique stress up to scaling. To see the degeneracy of G it suffices to show that a stress of a generic realization is degenerate. Consider a generic realization of $(G+23,p)$, and take two stresses ω_1 and ω_2 in the copies of K_5 on $\{1,2,3,4,5\}$ and $\{1,2,3,6,7\}$, respectively. For the set of neighbours of vertex 1 in each copy of K_5, the equilibrium condition implies

$$\omega_1 \circ p(\{4,5\}) \in \mathrm{span}\{\omega_1 \circ p(\{2\}), \omega_1 \circ p(\{3\})\} = \mathrm{span}\{p(2) - p(1), p(3) - p(1)\}.$$

We regard ω_1 and ω_2 as stresses of $(G+23,p)$, and by scaling we can suppose $\omega_1(23) + \omega_2(23) = 0$. Then $\omega := \omega_1 + \omega_2$ is a stress of (G,p). We consider the 3-subpartition $\{\{2\},\{3\},\{4,5\}\}$ of $N_G(1)$. By

$$\omega \circ p(\{4,5\}) = \omega_1 \circ p(\{4,5\}) \in \mathrm{span}\{p(2) - p(1), p(3) - p(1)\} = \mathrm{span}\{\omega \circ p(\{2\}), \omega \circ p(\{3\})\},$$

we conclude that ω is degenerate with respect to this subpartition. This example also shows that the so-called *coning* operation does not preserve nondegeneracy.

We however conjecture the following:

Conjecture 17 *Every globally rigid graph G in $\mathbb{R}^d$ is nondegenerate in $\mathbb{R}^d$.*

The truth of Conjecture 17 would imply the truth of the vertex splitting conjecture (Conjecture 10) by Theorem 19.

The following technical lemma is worth mentioning.

Lemma 18 *If G is redundantly rigid in $\mathbb{R}^d$ with maximum degree at most $d+2$ then G is nondegenerate in $\mathbb{R}^d$.*

Thus $K_{5,5}$ is an example of a non-globally rigid 4-connected M-circuit that is nondegenerate in $\mathbb{R}^3$ (c.f. [6]).

An important unsolved problem is to prove that nondegeneracy in $\mathbb{R}^d$ is preserved by the d-dimensional 1-extension operation and by edge-addition.

[1] A *k-subpartition* of a set A is a partition of a subset $B \subseteq A$ into k non-empty sets. We say that a subpartition is *proper* if B is a non-empty proper subset of A.

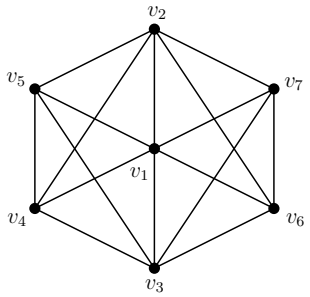


Figure 3: A graph that is M-connected but is degenerate in $\mathbb{R}^3$.

3.3 Proof of Theorem 11

Theorem 11 follows from Theorem 15, Lemma 18, and the following our main technical observation.

Theorem 19 *Let G be obtained from H by a nontrivial vertex splitting at v_1 with respect to partition $\{U_{01}, U_0, U_1\}$ of $N_G(v_1)$. Denote $U_{01} = \{u_1, \ldots, u_{d-1}\}$. Suppose that a generic framework (H, p) in $\mathbb{R}^d$ admits a full rank stress ω. Then*

(1) If ω is not degenerate with respect to $\{\{u_1\}, \ldots, \{u_{d-1}\}, U_0\}$, then some generic framework (G, p') admits a full rank stress.

(2) Moreover, if ω is nondegenerate, then (G, p') admits a full rank nondegenerate stress.

The proof is based on an idea for proving that Colin de Verdière graph parameter [3, 4] is minor-monotone by van der Holst, Lovász, and Schrijver [14]. It uses the transversality of the set of constant rank symmetric matrices and the set of weighted Laplacian of G. It turns out that the transversality corresponds in our context is equivalent to a conic condition for edge directions introduced by Connelly [5, 7].

By using the same proof technique we can prove that a 4-connected braced triangulation has an even stronger property.

Theorem 20 *Suppose that G can be obtained from K_{d+2} by a sequence of nontrivial vertex splitting operations. Then for any pair of nonnegative integers a and b with $a + b = n - d - 1$ there is a generic framework (G, p) in $\mathbb{R}^d$ that admits a stress matrix with inertia $(a, b, d + 1)$[2].*

References

[1] L. Asimow and B. Roth, The rigidity of graphs. *Trans. Amer. Math. Soc.*, 245:279–289, 1978.

[2] A.L. Cauchy, Sur les polygones et polyedres, second memoire, *J. Ecole Polytechnique*, 1813.

[3] Y. Colin de Verdière, Sur un nouvel invariant des graphes et un critère de planarité, *J. Comb. Theory, Ser. B*, 50:11-21, 1990.

[4] Y. Colin de Verdière, On a new graph invariant and a criterion of planarity, in: N. Robertson, P. Seymour (Eds.), *Graph Structure Theory, Contemporary Mathematics*, vol. 147, American Mathematical Society, Providence, RI, 1993, pp. 137–147.

[5] R. Connelly, Rigidity and energy, *Invent. Math.* 66:11–33, 1982.

[2] For a symmetric matrix S of size n whose numbers of positive and negative eigenvalues (with multiplicity) are a and b, respectively, its *inertia* is defined as triple $(a, b, n - a - b)$.

[6] R. Connelly, On generic global rigidity, in *Applied Geometry and Discrete Mathematics*, DIMACS Ser. Discrete Math, Theoret. Comput. Sci. 4, AMS, 1991, pp 147-155.

[7] R. Connelly, Generic global rigidity, *Discrete Comp. Geometry* 33 (2005), pp 549-563.

[8] R. Connelly, Questions, conjectures and remarks on globally rigid tensegrities, manuscript, November 2009.

[9] R. Connelly and W. Whiteley, Global rigidity: the effect of coning, *Discrete Comp. Geometry*, Volume 43, Number 4, 717-735 (2010).

[10] J. Cruickshank, D. Kitson, and S. Power, The generic rigidity of triangulated spheres with blocks and holes, *J. Combinatorial Theory, Series B*, Vol. 122, 2017, pp. 550-577.

[11] H. Gluck, Almost all simply connected closed surfaces are rigid. In *Geometric Topology*, volume 438 of *Lecture Notes in Mathematics*, pages 225–239. Springer-Verlag, 1975.

[12] S. Gortler, A. Healy, and D. Thurston, Characterizing generic global rigidity, *American Journal of Mathematics*, Volume 132, Number 4, August 2010, pp. 897-939.

[13] B. Hendrickson, Conditions for unique graph realizations, *SIAM J. Comput* 21 (1992), pp 65-84.

[14] H. van der Holst, L. Lovász, and A. Schrijver, The Colin de Verdière parameter. Graph theory and combinatorial biology (Balatonlelle, 1996), 29–85, Bolyai Soc. Math. Stud., 7, János Bolyai Math. Soc., Budapest, 1999.

[15] B. Jackson and T. Jordán, Connected rigidity matroids and unique realization graphs, *J. Combin. Theory Ser. B* 94, 2005, pp 1-29.

[16] B. Jackson and T. Jordán, Graph theoretic techniques in the analysis of uniquely localizable sensor networks, in: *Localization algorithms and strategies for wireless sensor networks*, G. Mao, B. Fidan (eds), IGI Global, 2009, pp. 146-173.

[17] T. Jordán, C. Király, and S. Tanigawa, Generic global rigidity of body-hinge frameworks, *J. Combinatorial Theory, Ser. B.*, Vol. 117, 59-76, 2016.

[18] T. Jordán and Z. Szabadka, Operations preserving the global rigidity of graphs and frameworks in the plane, *Computational Geometry*, 42 (2009) 511-521.

[19] W. Whiteley, Infinitesimally rigid polyhedra. II: Modified spherical frameworks, *Trans. Amer. Math. Soc.* 306, No. 1, 115-139, 1988.

[20] W. Whiteley, Vertex splitting in isostatic frameworks, *Structural Topology* 16 (1991), pp 23-30.

[21] W. Whiteley, Some matroids from discrete applied geometry, in Matroid theory (J.E. Bonin, J.G. Oxley and B. Servatius eds., Seattle, WA, 1995), Contemp. Math., 197, Amer. Math. Soc., Providence, RI, 1996, 171–311.

Practical Algorithms and Models for Evacuation Problems

Naoyuki Kamiyama[1]

Kyushu University
JST, PRESTO
kamiyama@imi.kyushu-u.ac.jp

Abstract: In this talk, we consider mathematical models for evacuation planning based on dynamic networks. A dynamic network is a directed graph in which arcs have capacities and transit times. The evacuation problem is one of the most fundamental problems in dynamic networks. The goal of this problem is to find a minimum time limit T such that we can send all supplies to the sink vertex within T. In the first half of this talk, we consider a practically faster algorithm for the evacuation problem based on time-expanded networks. More precisely, we first propose a theoretical algorithm for the evacuation problem by using a parametric submodular function minimization algorithm, and then we give a practical algorithm based on this theoretical algorithm. In the second half of this talk, we consider variants of the evacuation problem for modeling an emergent situation in which people can evacuate on foot or by car. The goal of this problem is to organize such a mixed evacuation so that an efficient evacuation can be achieved. For this problem, we give a polynomial-time algorithm for some special cases, and hardness results for variants of the mixed evacuation problem with integer constraints. Furthermore, we apply our model to the case study in Japan.

Keywords: dynamic network flow; evacuation problem; submodular function

1 Introduction

A dynamic network introduced by Ford and Fulkerson [6, 7] is a directed graph in which arcs have capacities and transit times (see, e.g., [18] for a detailed introduction to dynamic networks). A dynamic network is frequently used for modeling evacuation situations [2, 3, 4, 19] (see [8] for a survey of modeling based on dynamic networks). One of the most fundamental problems in dynamic networks is the evacuation problem. In this problem, we are given a dynamic network with a single sink vertex. Furthermore, we are given supplies for all vertices except the sink vertex. Then, the goal of this problem is to find a minimum time limit T such that we can send all supplies to the sink vertex within T. The contents of this talk is summarized as follows.

- In the first half of this talk, we propose a practically faster algorithm for the evacuation problem based on time-expanded networks. More precisely, we first propose a theoretical algorithm for the evacuation problem by using a parametric submodular function minimization algorithm, and then we give a practical algorithm based on this theoretical algorithm. This part is based on the results in [12].
- In the second half of this talk, we consider variants of the evacuation problem for modeling an emergent situation in which people can evacuate on foot or by car. The goal of this problem is to organize such a mixed evacuation so that an efficient evacuation can be achieved. For this problem, we give a polynomial-time algorithm for some special cases, and hardness results for variants of the

[1]This research was supported by JST, PRESTO Grant Number JPMJPR14E1, Japan.

mixed evacuation problem with integer constraints. Furthermore, we apply our model to the case study in Japan. This part is based on the results in [9].

2 An algorithm for the evacuation problem based on parametric submodular function minimization

It is known [10] that we can solve the evacuation problem in polynomial time by using a submodular function minimization algorithm (e.g., [11, 17]) as a subroutine in the framework of the binary search or the parametric search [13] (see also [16]). However, in practical applications, an algorithm based on a time-expanded network introduced by Ford and Fulkerson [6, 7] is frequently used instead of a submodular function minimization algorithm because an algorithm based on a time-expanded network can be easily implemented. Furthermore, an algorithm based on a time-expand network has the merit that we can easily find not only the minimum evacuation time but also a flow itself. Of course, there is a disadvantage of algorithms based on time-expanded networks. The size of a time-expanded network is very large, and thus the time required to solve the maximum flow problem in a time-expanded network is very long. Since the binary search framework is usually adopted for computing the optimal solution of the evacuation problem, the number of subproblems in a time-expanded network becomes large. The aim of this part is to propose a practically faster algorithm for the evacuation problem based on time-expanded networks by reducing the number of subproblems in a time-expanded network.

The contributions of this part are summarized as follows.

- We propose a theoretical algorithm on which our practical algorithm is based. More precisely, we prove that the evacuation problem can be solved in the same time complexity as that of the submodular function minimization algorithm of Orlin [15] by using a parametric submodular function minimization algorithm of Nagano [14].
- We propose a practical algorithm based on the theoretical algorithm. Our practical algorithm uses an algorithm computing a maximum flow in a time-expanded network in a place of a submodular function minimization algorithm in the theoretical algorithm. Furthermore, we compare our practical algorithm and an algorithm based on the binary search framework through computational experiments.

Fleischer and Skutella [5] proposed a theoretically faster approximation algorithm for the evacuation problem based on a time-expanded network. However, to the best of our knowledge, there does not exist a paper aiming to propose a practically faster exact algorithm for the evacuation problem based on a time-expanded network.

3 The mixed evacuation problem

The coastal area facing the Pacific Ocean in Japan ranging from Shizuoka prefecture to Miyazaki prefecture has a high risk of a tsunami. In particular, it is predicted that Nankai Trough Earthquake will occur with 70% probability within thirty years, and it will trigger a tsunami of the huge size which will quickly arrive at the coast (see, e.g., [1]). Based on several assumptions and estimated data, Wakayama prefecture recently designated several areas in which it is difficult for all people in the area to evacuate to safety places such as tsunami evacuation buildings before a tsunami arrives when Nankai Trough Earthquake occurs. For example, it is predicted that in Kushimoto town located at the south end of the main land of Japan, a tsunami arrives at earliest within ten minutes. One of assumptions the prefecture used is that the evacuation is done only by walking. In principle, it used to be not allowed to use cars for evacuation because the usage of cars in such an emergent situation may block evacuation of pedestrians which was observed at the time of Tohoku-Pacific Ocean Earthquake. However, if it is allowed to use cars and the smooth evacuation by car is organized, then the evacuation completion time may be shortened. The aim of this part is to propose a mathematical model for making such a good "mixed" evacuation plan.

In this part, we introduce a variant of the quickest transshipment problem called the mixed evacuation problem. This problem models an emergent situation in which people can evacuate on foot or by car. The goal of this problem is to organize such a mixed evacuation so that an efficient evacuation can be achieved. In the first half of this part, we consider the mixed evacuation problem from the theoretical viewpoint. First we prove that if the number of sources and sinks is at most $C \log_2 n$ (n is the number of vertices) for some constant C, then mixed evacuation problem can be solved in polynomial time. In addition, we consider variants of the mixed evacuation problem with integer constraints, In the second half of this part, we study the mixed evacuation problem from the practical viewpoint. We apply our model to the case study in Japan. More precisely, we apply our model for Minabe town in Wakayama prefecture, which was designated as a city in which safe evacuation from a tsunami is difficult when Nankai Trough Earthquake occurs.

References

[1] http://www.jishin.go.jp/main/choukihyoka/ichiran.pdf.

[2] D. Dressler, G. Flötteröd, G. Lämmel, K. Nagel, and M. Skutella. Optimal evacuation solutions for large-scale scenarios. In *Operations Research Proceedings 2010*, pages 239–244, 2010.

[3] D. Dressler, M. Groß, J.-P. Kappmeier, T. Kelter, J. Kulbatzki, D. Plümpe, G. Schlechter, M. Schmidt, M. Skutella, and S. Temme. On the use of network flow techniques for assigning evacuees to exits. *Procedia Engineering*, 3:205–215, 2010.

[4] C. Even, V. Pillac, and P. Van Hentenryck. Convergent plans for large-scale evacuations. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 1121–1127, 2015.

[5] L. Fleischer and M. Skutella. Quickest flows over time. *SIAM Journal on Computing*, 36(6):1600–1630, 2007.

[6] L. R. Ford, Jr. and D. R. Fulkerson. Constructing maximal dynamic flows from static flows. *Operations Research*, 6(3):419–433, 1958.

[7] L. R. Ford, Jr. and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[8] H. W. Hamacher and S. A. Tjandra. Mathematical modelling of evacuation problem: state of the art. In M. Schreckenberg and S. D. Sharma, editors, *Pedestrian and Evacuation Dynamics*, pages 227–266. Springer, 2002.

[9] Y. Hanawa, Y. Higashikawa, N. Kamiyama, N. Katoh, and A. Takizawa. The mixed evacuation problem. In *Proceedings of the 10th Annual International Conference on Combinatorial Optimization and Applications*, volume 10043 of *Lecture Notes in Computer Science*, pages 18–32, 2016.

[10] B. Hoppe and É. Tardos. The quickest transshipment problem. *Mathematics of Operations Researh*, 25(1):36–62, 2000.

[11] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777, 2001.

[12] N. Kamiyama. An algorithm for the evacuation problem based on parametric submodular function minimization. Technical Report MI Preprint Series 2016-14, Kyushu University, 2016.

[13] N. Megiddo. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4(4):414–424, 1979.

[14] K. Nagano. A faster parametric submodular function minimization algorithm and applications. Technical Report METR 2007-43, The University of Tokyo, 2007.

[15] J. B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

[16] M. Schlöter and M. Skutella. Fast and memory-efficient algorithms for evacuation problems. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 821–840, 2017.

[17] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[18] M. Skutella. An introduction to network flows over time. In W. Cook, L. Lovasz, and J. Vygen, editors, *Research Trends in Combinatorial Optimization*, pages 451–482. Springer, 2009.

[19] A. Takizawa, M. Inoue, and N. Katoh. An emergency evacuation planning model using the universally quickest flow. *The Review of Socionetwork Strategies*, 6(1):15–28, 2012.

Approximating Volume — Randomized vs. Deterministic

Shuji Kijima[1]

Graduate School of Information Science and
Electrical Engineering, Kyushu University,
744 Motooka, Nishi-ku, Fukuoka, Japan
JST PRESTO,
744 Motooka, Nishi-ku, Fukuoka, Japan
kijima@inf.kyushu-u.ac.jp

Abstract: Counting is a central topic in Combinatorics. It is also highly related to Probability and/or Computing. The topic of this talk is randomized and deterministic approximation of counting hard problems.

Keywords: #P-hard, FPRAS, FPTAS, #BIS, log-supermodular, Tutte Polynomial

1 Introduction

Counting is a central topic in Combinatorics; the number of permutations is given by a factorial, the number of combinations is given by a binomial coefficient, the number of Dyck paths, parentheses, binary trees are associated with Catalan numbers, the number of spanning trees is efficiently calculated by the Matrix tree theorem, etc. Counting is also highly related to probability and computation, in particular, integrations [16], and approximating volume is an interesting topic in Combinatorics, Probability and Computing.

2 Randomized Approximate Counting of Self-avoiding Walks

This section briefly shows an example of randomized approximate counting based on the Markov chain Monte Carlo (MCMC) method.

Let Ω_n denote the set of *simple* paths from the north-west corner (s) to the south-east corner (t) of the $n \times n$ grid. Figure 1 shows a 5×5 grid. It is easy to check that $|\Omega_1| = 2$, $|\Omega_2| = 12$, $|\Omega_3| = 184$, [28, 27]. In fact, the number is known up to $n = 26$, and $|\Omega_{26}| = 1736993158627927293117544042123649890037222958828814060466370372091034241327613476278921819349800610708229622314338049134829002672193112962770873889085390810890639 6 \simeq 1.74 \times 10^{164}$ [27], where it took about one week to obtain the value by a computer server with 80 cores, using an algorithm based on the ZDD technique by [29, 15].

Using a standard MCMC technique, we can approximately count Ω_n for large values of n, such as $n = 100$. Let L denote the length of the longest path, that is $L = (n+1)^2 - 1$ if n is odd, otherwise $L = (n+1)^2 - 2$. Let Ξ_k ($k = 2n, 2n+2, \ldots, L$) denote the set of simple paths with length at most k in Ω_n. Then it is easy to observe that

$$|\Omega_n| = |\Xi_L| = \frac{|\Xi_L|}{|\Xi_{L-2}|} \cdot \frac{|\Xi_{L-2}|}{|\Xi_{L-4}|} \cdot \cdots \cdot \frac{|\Xi_{2n+2}|}{|\Xi_{2n}|} \cdot |\Xi_{2n}|$$

[1]This research was/is partly supported by MEXT KAKENHI Grant Number 24106005, JSPS KAKENHI Grant Number 25700002, JST PRESTO Grant Number JPMJPR16E4.

Figure 1: Counting self-avoiding walks on the grid (see also the youtube animation [28])

holds, where $|\Xi_{2n}| = \binom{2n}{n}$ holds, in fact. Suppose we have a uniform random sampler for Ξ_k for $k = 2n+2, 2n+4, \ldots, L$, then we can approximate $|\Xi_{k-2}|/|\Xi_k|$ by a Monte Carlo method. For the purpose, we design a Markov chain whose stationary distribution is uniform over Ξ_k.

To describe the idea of Markov chain, we introduce a representation of a simple path by coloring of cells of the grid. Consider a simple path p from t to s out of the grid region. Then any simple path from s to t on a grid and the path p form a closed curve C. Let us assign black to each cell surrounded by C, and let us assign white to each cell outside of C, then we obtain a coloring of the cells. It is not difficult to observe that the coloring is bijective to Ω_n. Now, the Markov chain is easy to give: choose a cell uniformly at random, and flip the color of the cell if the change of the state is valid, otherwise keep the color of all cells. By a standard argument on the MCMC, we can check that the Markov chain has a unique limit distribution, and it is uniform over Ξ_k.

Implementing the above algorithm, we obtain the approximate value $|\Omega_{100}| \simeq 6.07 \times 10^{2415}$. The computation took about 24 hours on a standard computer. We remark that the cell-coloring representation implies a trivial upper bound 2^{n^2} of $|\Omega_n|$. We conjecture that $|\Omega_n|$ is lower bounded by $\sqrt{3}^{n^2}$ [23]. It is not difficult to show a lower bound $\sqrt[3]{3}^{n^2}$, meaning that $\log(|\Omega_n|)$ grows quadratic to n. We also refer to the result [6] by Bousquet-Mélou et al. It is another future work to prove a Markov chain with the uniform stationary distribution over Ξ_k mixes in poly(n) time.

3 Deterministic Approximation of the Volume of a Polytope

A high dimensional volume is hard to compute, even for approximation. When an n-dimensional convex body is given by a *membership oracle*, no polynomial-time *deterministic* algorithm can approximate its volume within ratio $(n/\log n)^n$ [4, 13, 8].

Several *randomized* approximation techniques for #P-hard problems have been developed, such as the Markov chain Monte Carlo method. For the volume computation of a general convex body given by a membership oracle in the n dimensional space, Dyer, Frieze and Kannan [11] gave the first *fully polynomial-time randomized approximation scheme* (*FPRAS*), giving a rapidly mixing Markov chain. In fact, the running time of the FPRAS is $\mathrm{O}^*(n^{23})$ where O^* ignores poly$(\log n)$ and $1/\epsilon$ terms. After several improvements, Lovász and Vempala [21] improved the time complexity to $\mathrm{O}^*(n^4)$, and recently Cousins and Vempala [7] gave an $\mathrm{O}^*(n^3)$-time algorithm.

In contrast, it is a major challenge to design *deterministic* approximation algorithms for #P-hard problems, and not many results seem to be known. A remarkable progress is the *correlation decay* argument due to Weitz [26]; he designed a *fully polynomial time approximation scheme* (*FPTAS*) for counting independent sets in graphs whose maximum degree is at least 5. A similar technique is independently presented by Bandyopadhyay and Gamarnik [3], and there are several recent developments on the technique. For counting 0-1 knapsack solutions, Gopalan, Klivans and Meka [14], and Stefankovic, Vempala and Vigoda [24] gave deterministic approximation algorithms based on the dynamic programming, in a

similar way to a simple random sampling algorithm by Dyer [9].

3.1 An FPTAS for the volume of 0-1 a knapsack polytope

Motivated by a new technique of designing an FPTAS for #P-hard problems, Ando and Kijima [1] were concerned with the volume of 0-1 knapsack polytope, and gave a *deterministic* approximation; Given a positive integer vector $\boldsymbol{a}^\top = (a_1, \ldots, a_n) \in \mathbb{Z}^n_{>0}$ and a positive integer $b \in \mathbb{Z}_{>0}$, the 0-1 *knapsack polytope* is given by

$$K \stackrel{\text{def}}{=} \{\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n \mid \boldsymbol{a}^\top \boldsymbol{x} \leq b,\ 0 \leq x_i \leq 1\ (i = 1, \ldots, n)\}.$$

Computing the *volume* of K is known to be #P-hard [10]. Remark that counting solutions corresponds to counting the integer points in K, and it is different from the volume computation, but closely related.

Theorem 1 ([1]) *For any ϵ $(0 < \epsilon \leq 1)$, there exists an $\mathrm{O}(n^3/\epsilon)$-time* deterministic *algorithm to approximate $\mathrm{Vol}(K)$ with approximation ratio $1 + \epsilon$.*

The algorithm is based on the classical *convolution*. The key technology of the paper is a development of the techniques for bounding approximation ratio, which are horizontal approximation and cone approximation. We omit the detail here, and see [1] for the detail.

3.2 An FPTAS for the volume of some $\mathcal{V}$-polytopes

$\mathcal{H}$-polytope and $\mathcal{V}$-polytope An $\mathcal{H}$-polyhedron is an intersection of finitely many closed half-spaces in $\mathbb{R}^n$. An $\mathcal{H}$-polytope is a bounded $\mathcal{H}$-polyhedron. A $\mathcal{V}$-polytope is a convex hull of a finite point set in $\mathbb{R}^n$ [22]. From the view point of computational complexity, a major difference between an $\mathcal{H}$-polytope and a $\mathcal{V}$-polytope is the measure of their 'input size.' An $\mathcal{H}$-polytope given by linear inequalities defining half-spaces may have vertices exponentially many to the number of the inequalities, e.g., an n-dimensional hypercube is given by $2n$ linear inequalities as an $\mathcal{H}$-polytope, and has 2^n vertices. In contrast, a $\mathcal{V}$-polytope given by a point set may have facets exponentially many to the number of vertices, e.g., an n-dimensional cross-polytope (that is an L_1-ball, in fact) is given by a set of $2n$ points as a $\mathcal{V}$-polytope, and it has 2^n facets.

There are many interesting properties, that are known, or unknown, between $\mathcal{H}$-polytope and $\mathcal{V}$-polytope [22]. A membership query is polynomial time for both $\mathcal{H}$-polytope and $\mathcal{V}$-polytope. It is still unknown about the complexity of a query if a given pair of $\mathcal{V}$-polytope and $\mathcal{H}$-polytope are identical. Linear programming (LP) on a $\mathcal{V}$-polytope is trivially polynomial time since it is sufficient to check the objective value of all vertices and hence LP is usually concerned with an $\mathcal{H}$-polytope.

The volume of some $\mathcal{V}$-polytopes Motivated by a hardness of the volume computation of a $\mathcal{V}$-polytope, Khachiyan [17] is concerned with the following $\mathcal{V}$-polytope: Suppose a vector $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{Z}^n_{\geq 0}$ is given, where without loss of generality we may assume that $a_1 \geq a_2 \geq \cdots \geq a_n$. Then let

$$P_{\boldsymbol{a}} \stackrel{\text{def}}{=} \mathrm{conv}\,\{\pm\boldsymbol{e}_1, \ldots, \pm\boldsymbol{e}_n, \boldsymbol{a}\} \tag{1}$$

where $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ are the standard basis vectors in $\mathbb{R}^n$. Khachiyan [17] showed that computing $\mathrm{Vol}(P_{\boldsymbol{a}})$ is $\#P$-hard. The Polytope $P_{\boldsymbol{a}}$ given by (1) is somehow (geometric) "dual polytope" of 0-1 knapsack polytope (see e.g., [22, 18, 2]). However, we do not know any (efficient) technique to translate from the volume of a polytope to that of its dual polytope.

Motivated by a development of techniques for a *deterministic* approximation of the volumes of $\mathcal{V}$-polytopes, Ando and Kijima [2] gave a deterministic approximation of the volume of $P_{\boldsymbol{a}}$ given by (1).

Theorem 2 ([2]) *For any ϵ $(0 < \epsilon < 1)$, there exists a* deterministic *algorithm that outputs a value $\widehat{V}$ satisfying $(1-\epsilon)\mathrm{Vol}(P_{\boldsymbol{a}}) \leq \widehat{V} \leq (1+\epsilon)\mathrm{Vol}(P_{\boldsymbol{a}})$ in $\mathrm{O}(n^{10}\epsilon^{-6})$ time.*

As far as we know, this is the first result on designing an FPTAS for the volume of a $\mathcal{V}$-polytope which is known to be #P-hard.

In fact, [2] reduces the problem to compute the volume of the intersection of two cross polytopes. A *cross-polytope* $C(\boldsymbol{c}, r)$ of radius $r \in \mathbb{R}_{>0}$ centered at $\boldsymbol{c} \in \mathbb{R}^n$ is given by

$$C(\boldsymbol{c}, r) \stackrel{\text{def}}{=} \text{conv}\{\boldsymbol{c} \pm r\boldsymbol{e}_i \ i = 1, \ldots, n\}$$

where $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ are the standard basis vectors in $\mathbb{R}^n$. Clearly, $C(\boldsymbol{c}, r)$ has $2n$ vertices. In fact, $C(\boldsymbol{c}, r)$ is an L_1-ball in $\mathbb{R}^n$ described by

$$\begin{aligned} C(\boldsymbol{c}, r) &= \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x} - \boldsymbol{c}\|_1 \leq r\} \\ &= \{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \boldsymbol{x} - \boldsymbol{c}, \boldsymbol{\sigma} \rangle \leq r \ (\forall \boldsymbol{\sigma} \in \{-1, 1\}^n)\} \end{aligned}$$

where $\|\boldsymbol{u}\|_1 = \sum_{i=1}^n |u_i|$ for $\boldsymbol{u} = (u_1, \ldots, u_n) \in \mathbb{R}^n$ and $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{i=1}^n u_i v_i$ for $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$. Note that $C(\boldsymbol{c}, r)$ has 2^n facets. It is not difficult to see that the volume of a cross-polytope in n-dimension is

$$\text{Vol}(C(\boldsymbol{c}, r)) = \frac{2^n}{n!} r^n$$

for any $r \geq 0$ and $\boldsymbol{c} \in \mathbb{R}^n$, where $\text{Vol}(S)$ for $S \subseteq \mathbb{R}^n$ denotes the (n-dimensional) volume of S.

Then, [2] provided an FPTAS for the volume of an intersection of two cross polytopes.

Theorem 3 ([2]) *For any δ $(0 < \delta < 1)$, there exists a* deterministic *algorithm which outputs a value Z satisfying $\text{Vol}(C(\mathbf{0}, 1) \cap C(\boldsymbol{c}, r)) \leq Z \leq (1+\delta)\text{Vol}(C(\mathbf{0}, 1) \cap C(\boldsymbol{c}, r))$ for any input $\boldsymbol{c} \geq \mathbf{0}$ and r $(0 < r \leq 1)$ satisfying $\|\boldsymbol{c}\|_1 \leq r$, and runs in $O(n^7 \delta^{-3})$ time.*

The assumption that $\|\boldsymbol{c}\|_1 \leq r$ implies both centers $\mathbf{0}$ and $\boldsymbol{c}$ are contained in the intersection $C(\mathbf{0}, 1) \cap C(\boldsymbol{c}, r)$. Note that the assumption does not harm to our main goal Theorem 2. Furthermore, $\text{Vol}(C(\mathbf{0}, 1) \cap C(\boldsymbol{c}, r))$ remains $\#P$-hard even on the assumption.

Theorem 4 ([2]) *Given a vector $\boldsymbol{c} \in \mathbb{Z}_{>0}^n$ and integers $r_1, r_2 \in \mathbb{Z}_{>0}$, computing the volume of $C(\mathbf{0}, r_1) \cap C(\boldsymbol{c}, r_2)$ is #P-hard, even when each cross-polytopes contains the center of the other one, i.e., $\mathbf{0} \in C(\boldsymbol{c}, r_2)$ and $\boldsymbol{c} \in C(\mathbf{0}, r_1)$.*

See [2] for more detail. The complexity, such as FPTAS, #P-hardness, of computing volume of the intersection of two crosspolytopes unless both centers are contained in their intersection is unknown.

4 The Number of Ideals and Linear Extensions of a Poset

#BIS is a problem to count the number of independent sets in a given bipartite graph $G = (U, V; E)$. It is a major open problem if #BIS has an FPRAS or not. Related to #BIS, this section is concerned with counting ideals of a partially ordered set (poset), and linear extensions of a poset.

4.1 Counting ideals

Let E be a partially ordered set associated with $\preceq$. An *ideal* (or *down set*) of the poset $(E, \preceq)$ is a subset $F \subseteq E$ such that any $x \in F$ if there is $y \in E$ and $x \preceq y$. Ideals forms a distributive lattice, and any distributive lattice is represented by a family of ideals of a partially ordered set. due to the celebrated *Birkhoff's representation theorem*. #Ideal is a problem to count the number of ideals of a poset.

Theorem 5 ([12]) *#BIS has an FPRAS if and only if #Ideal has an FPRAS.*

For convenience, a problem is #BIS-hard if it is reduced to #BIS in an approximate preserving way, i.e., if the problem has an polynomial time approximation scheme then so does #BIS. Thus, #Ideals is #BIS-hard.

Stable matching is a topic highly related to, or a killer application of, ideals of a poset. It is a well known fact that the set of stable matchings of any instance of the stable marriage problem forms a distributive lattice under women's (or men's) preference. It is also known that any finite distributive lattice has an instance of the stable marriage problem whose lattice of stable matchings are isomorphic to the given distributive lattice [5]. The conflict between men and women is an important issue in a practical application of stable matching, and a *median stable matching*, devised by Teo and Sethuraman [25], is a fair stable matching between men and women. Nemoto and Kijima [20] showed that it is also #BIS-hard.

4.2 Counting linear extensions

Let $(E, \preceq)$ be a poset of order n. An *linear extension* of $(E, \preceq)$ is an entire sequence $e_1, e_2, \ldots, e_n$ of E such that $i < j$ hold if $e_i \preceq e_j$. #LinEx is a problem to count the number of linear extensions of a poset. Some FPRAS are known for #LinEx. Here, we briefly mention to an FPRAS based on a volume computation of an order polytope.

Given a poset $(E, \preceq)$, the *order polytope* associated with E is given by

$$P(E) \stackrel{\text{def}}{=} \{\boldsymbol{x} \in [0,1] \mid x_i \leq x_j \text{ if } i \preceq j \ (i,j \in E)\}.$$

We can observe that n-dimensional the volume of $P(E)$ is equal to the number of linear extensions of E. Using an FPRAS for a general convex body [11, 21, 7], we can approximately count the number of linear extensions. It is not known if there is an FPTAS for #LinEx.

We remark that the the number of vertices of $P(E)$ is equal to the number of ideals of E, i.e., counting the number of vertices of a order polytope is #BIS-hard. As a related topic, sampling from log-supermodular distribution is also #BIS-hard [19]. Sampling from log super/submodular distribution is related to Tutte polynomial, and it contains another major open problem if there is an FPRAS for the number of forests of a graph.

Acknowledgement

The author would like to thank all the collaborators of the researches mentioned in this paper.

References

[1] E. Ando and S. Kijima, An FPTAS for the volume computation of 0-1 knapsack polytopes based on approximate convolution, *Algorithmica* **76** (2016), 1245–1263.

[2] E. Ando and S. Kijima, An FPTAS for the volume of a $\mathcal{V}$-polytope —it is hard to compute the volume of the intersection of two cross-polytopes, *arXiv:1607.06173*, 2016.

[3] A. Bandyopadhyay and D. Gamarnik, Counting without sampling: asymptotics of the log-partition function for certain statistical physics models, *Random Structures and Algorithms*, **33** (2008), 452–479.

[4] I. Bárány and Z. Füredi, Computing the volume is difficult, *Discrete Computational Geometry*, **2** (1987), 319–326.

[5] C. Blair, Every finite distributive lattice is a set of stable matchings, *Journal of Combinatorial Theory, Series A*, **37** (1984), 353–356.

[6] M, Bousquet-Mélou, A. J. Guttmann and I. Jensen, Self-avoiding walks crossing a square, *Journal of Physics A: Mathematical and General*,**38** (2005), 9159–9182.

[7] B. Cousins and S. Vempala, Bypassing KLS: Gaussian cooling and an $O^*(n^3)$ volume algorithm, *Proceedings of STOC 2015*, 539–548.

[8] D. Dadush and S. Vempala, Near-optimal deterministic algorithms for volume computation via M-ellipsoids, *Proc. Natl. Acad. Sci.* USA 2013 Nov 26, **110**, 19237–19245.

[9] M. Dyer, Approximate counting by dynamic programming, *Proc. of STOC 2003*, 693–699.

[10] M. Dyer and A. Frieze, On the complexity of computing the volume of a polyhedron, *SIAM Journal on Computing*, **17** (1988), 967–974.

[11] M. Dyer, A. Frieze and R. Kannan, A random polynomial-time algorithm for approximating the volume of convex bodies, *Journal of the Association for Computing Machinery*, **38** (1991), 1–17.

[12] M. Dyer, L. A. Goldberg, C. Greenhill and M. Jerrum, The relative complexity of approximate counting problems, *Algorithmica*, **38** (2003), 471–500.

[13] G. Elekes, A geometric inequality and the complexity of computing volume, *Discrete Computational Geometry*, **1** (1986), 289–292.

[14] P. Gopalan, A. Klivans and R. Meka, Polynomial-time approximation schemes for knapsack and related counting problems using branching programs, arXiv:1008.3187v1, 2010.

[15] H. Iwashita, Y. Nakazawa, J. Kawahara, T. Uno and S. Minato, Efficient computation of the number of paths in a grid graph with minimal perfect hash functions *Hokkaido University*, Division of Computer Science, TCS Technical Reports, TCS-TR-A-13-64, Apr. 2013.

[16] M. Jerrum, L, G. Valiant, V. V. Vazirani, Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, **43** (1986), 169–188.

[17] L. Khachiyan, The problem of computing the volume of polytopes is $\#P$-hard, *Uspekhi Mat. Nauk.*, **44** (1989), 199–200.

[18] L. Khachiyan, Complexity of polytope volume computation; *In New Trends in Discrete and Computational Geometry*, (ed by J. Pach), Springer, Berlin, 1993, pp.91-101.

[19] S. Kijima, Sampling from log-super/submodular distributions, *Proceedings of the 7th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications*, (2011), 227–236.

[20] S. Kijima and T. Nemoto, On randomized approximation for finding a level ideal of a poset and the generalized median stable matchings, *Mathematics of Operations Research*, **37** (2012), 356–371.

[21] L. Lovász and S. Vempala, Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm, *Journal of Computer and System Sciences*, **72** (2006), 392–417.

[22] J. Matoušek, *Lectures on Discrete Geometry*, Springer, 2002.

[23] Y. Shibata, Y. Yamauchi, S. Kijima and M. Yamashita, Randomized approximate counting of self-avoiding walks, http://tcslab.csce.kyushu-u.ac.jp/˜kijima/posters/Minato16kijima.pdf

[24] D. Štefankovič, S. Vempala and E. Vigoda, A deterministic polynomial-time approximation scheme for counting knapsack solutions, *SIAM Journal on Computing*, **41** (2012), 356–366.

[25] C. P. Teo and J. Sethuraman, The geometry of fractional stable matchings and its applications, *Mathematics of Operations Research*, **23** (1998), 874–891.

[26] D. Weitz, Counting independent sets up to the tree threshold, *Proc. STOC 2006*, 140–149.

[27] https://oeis.org/A007764

[28] MiraikanChannel. Time with class! Lets count! http://www.youtube.com/watch?v=Q4gTV4r0zRs.

[29] https://h50146.www5.hpe.com/products/servers/proliant/whitepaper/pdfs/WhitePaper_HokkaidoUni_ST.pdf

Stable project allocation under distributional constraints

Kolos Csaba Ágoston

Department of Operations Research and Actuarial Sciences
Corvinus University of Budapest
H-1093, Fővám tér 13-15., Budapest, Hungary
kolos.agoston@uni-corvinus.hu

Péter Biró[1]

Institute of Economics, Research Centre for Economic and Regional Studies,
Hungarian Academy of Sciences, H-1112, Budaörsi út 45, Budapest, Hungary, and
Department of Operations Research and Actuarial Sciences, Corvinus University of Budapest
peter.biro@krtk.mta.hu

Richárd Szántó

Department of Decision Sciences
Corvinus University of Budapest
H-1093, Fővám tér 13-15., Budapest, Hungary
richard.szanto@uni-corvinus.hu

Abstract: In a two-sided matching market when agents on both sides have preferences the stability of the solution is typically the most important requirement. However, we may also face some distributional constraints with regard to the minimum number of assignees or the distribution of the assignees according to their types. These two kind of requirements can be challenging to reconcile in practice. Our research is motivated by two real applications, a project allocation problem and a workshop assignment problem, both involving some distributional constraints. We used integer programming techniques to find reasonably good solutions with regard to the stability and the distributional constraints. Our approach can be useful in a variety of different applications, such as resident allocation with lower quotas, controlled school choice or college admissions with affirmative action.

Keywords: stable matching, two-sided markets, project allocation, linear programming, multi-criteria decision making

1 Introduction

Centralised matching scheme has been used since 1952 in the US to allocate junior doctors to hospitals [29]. Later, the same technology has been used in school choice programs in large cities, such as New York [3] and Boston [4]. Similar schemes have been established in Europe for university admissions and school choice as well. For instance, in Hungary both the secondary school and the higher education admission schemes are organised nationwide, see [9] and [10], respectively. In the above mentioned applications it is common that the preferences of the applicants and the rankings of the parties on the other side are collected by a central coordinator and a so-called stable allocation is computed based on the matching algorithm of Gale and Shapley [19]. Two-sided matching markets, and the above applications

[1]Research is supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2016-3/2016), and by OTKA grant no. K108673.

in particular, have been extensively studied in the last decades, see [32] and [26] for overviews from game theoretical and computational aspects, respectively.

This paper is motivated by two applications at the Corvinus University of Budapest. In the first application the task is to allocate students to projects in such a way that the number of students allocated to each project is between a lower and an upper quota. This is a natural requirement present in many applications, such as the Japanese resident allocation scheme [22, 23, 20]. Furthermore, there are also separate lower bounds on the number of foreign students assigned to each company. In the second application the goal is to assign students to companies for solving case studies in a conference, and here again some distributional constraints are imposed with regard to the total number of local, European and other students selected.

We decided to investigate the integer programming techniques for solving these problems motivated by both applications. We had at least three reasons for choosing this technique. The first is that with IP formulations we can easily encode those distributional requirements that the organisers requested, so this solution method is robust to accommodate special features. The second reason is that the computational problem became NP-hard as the companies submitted lists with ties. Using ties in the ranking was by our recommendation to the companies, because ties give us more flexibility when finding a stable solution under the distributional constraints. We describe this issue more in detail shortly. Finally, our third reason for choosing IP techniques was that it facilitates multi-objective optimisation, e.g. finding a most-stable solution if a stable solution does not exist under the strict distributional constraints.

The usage of integer programming techniques for solving two-sided stable matching problems is very rare in the applications, and the theoretical studies on this topic have only started very recently. The reason is that the problems are relatively large in most applications, and the Gale-Shapley type heuristics are usually able to find stable solutions, even in potentially challenging cases. A classical example is the resident allocation problem with couples, which has been present in the US application for decades, and it is still solved by the Roth-Peranson heuristic [31]. The underlying matching problem is NP-hard [28], but heuristic solutions are quite successful in practice, see also [11] on the Scottish application. However, integer programming and constraints programming techniques have been developed very recently and they turned out to be powerful enough to solve large random instances [13], [14] and [15]. Similarly encouraging results have been obtained for some special college admission problems, which are present in the Hungarian higher education system. These special features also makes the problem NP-hard in general, but at least one of these challenging features, turned out to be solvable even in a real data involving more than 150,000 applicants [5]. Finally, the last paper that we highlight with regard to this topic deals with the problem of finding stable solutions in the presence of ties [25]. However, we are not aware of any papers that would study IP techniques for the problem of distributional constraints.

Distributional constraints are present in many two-sided matching markets. In the Japanese resident allocation the government wants to ensure that the doctors [22, 23, 20] are evenly distributed across the country, and to achieve this they imposed lower quotas on the number of doctors allocated in each region. Distributional objectives can also appear in school choice programs [2, 12, 17], where the decision makers want to control the socio-ethnical distribution of the students. Furthermore, the same kind of requirements are implemented in college admission schemes with affirmative action [1] such as the Brazilian college admission system [6] and the admission scheme to Indian engineering schools [7].

When stable solution does not exists for the strict distributional constraints then we either need to relax stability or to adjust the distributional constraints. In this study we will consider the trade-off between these two goals, and develop some reasonable solution concepts.

2 Definitions and preliminaries

Many-to-one stable matching markets have been defined in many context in the literature. In the classical college admissions problem by Gale and Shapley [19] the students are matched to colleges. In the computer science literature this problem setting is typically called Hospital / Residents problem (HR), due to the NRMP and other related applications. In our paper we will refer the two sets as *applicants*

$A = \{a_1, \ldots, a_n\}$ and *companies* $\{C = c_1, \ldots c_m\}$. Let u_j denote the upper quota of company c_j.

Regarding the preferences, we assume that the applicants provide strict rankings over the companies, but the companies may have ties in their rankings. This model is sometimes referred to as Hospital / Residents problem with Ties (HRT) in the computer science literature, see e.g. [26]. In our context, let r_{ij} denote the rank of company c_j in a_i's preference list, meaning that applicant a_i prefers c_j to c_k if and only if $r_{ij} < r_{ik}$. Let s_{ij} be an integer representing the score of a_i by company c_j, meaning that a_i is preferred over a_k by company c_j if $s_{ij} > s_{kj}$. Note that here two applicants may have the same score at a company, so $s_{ij} = s_{kj}$ is possible. Let $\bar{s}$ denote the maximum possible score at any company and let E be the set of applications. A *matching* is a subset of applications, where each applicant is assigned to at most one company and the number of assignees at each company is less than or equal to the upper quota. A matching is said to be *stable* if for any applicant-company pair not included in the matching either the applicant is matched to a more preferred company or the company filled its upper quota with applicants of the same or higher scores.

In the classical college admission problem, that we refer to as HR, a stable solution is guaranteed to exist, and the two-versions of the Gale-Shapley algorithm [19] find either a student-optimal or a college optimal solution, respectively. Furthermore, this algorithm can be implemented to run in linear time in the number of applications. Moreover, the student-proposing variant was also proved to be strategyproof for the students [29], which means that no student can ever get a better partner by submitting false preferences. Finally, the so-called Rural Hospitals Theorem [30] states that the same students are matched in every stable solution, the number of assignees does not vary across stable matchings for any college, and for the less popular colleges where the upper quota is not filled the set of assigned students is fixed.

When extending the classical college admission problem with the possibility of having ties in the colleges' rankings, that we referred to as an HRT instance, the existence of a stable solution is still guaranteed, since we can break the ties arbitrarily, and a stable solution for the strict preferences is also stable for the original ones. However, now the set of matched students and the size of the stable matchings can vary. Take just the following simple example: we have two applicants, a_1 and a_2 first applying to college c_1 with the same score and applicant a_2 also applies to college c_2 as her second choice. Here, if we break the tie at c_1 in favour of a_1 then we get the matching a_1c_1, a_2c_2, whilst if we break the tie in favour of a_2 then the resulting stable matching is a_2c_1 (thus a_1 is unmatched). The problem of finding a maximum size stable matching turned out to be NP-hard [27], and has been studied extensively in the computer science literature, see e.g. [26]. Note that when the objective of an application is to find a maximum size stable matchings, such as the Scottish resident allocation scheme [21], then the mechanism is not stategyproof. To see this, we just have to reconsider the above example, and assume that originally a_1 also found c_2 acceptable and would ranked it second, just like a_2. By removing c_2 from her list, a_1 is now guaranteed to get c_1 is the maximum size stable solution, however, for the original true preferences a_2 would have an equal chance to get her first choice c_1.

2.1 Introduction of lower quotas

In our first application the organisers of the project allocations wanted to ensure a minimum number of students for each company. Similar requirements have been imposed for the Japanese regions with regard to the number of residents allocated there. In our model, we introduce a lower quota l_j for each company c_j and we require that in a feasible matching the number of assignees at any company is between the lower and upper quotas. Stability is defined as before. We refer to the setting with strict preferences as Hospitals / Residents problem with Lower quotas (HRL) and the case with ties is referred to as Hospitals / Residents problem with Ties and Lower Quotas (HRTL).

Regarding HRL, the Rural Hospitals Theorem implies that the existence of a stable matching that obeys both the lower an upper quotas can be decided efficiently. This is because we just find one stable matching by considering the upper quotas only, and if the lower quotas are violated then there exists no stable solution under these distributional constraints. This problem can be still solved efficiently when the sets of companies have common lower and upper quotas in a laminar system, see [18].

However, the problem of deciding the existence of a stable matching for HRTL is NP-hard. To see this,

we just have to remark that the problem of finding a complete stable matching for HRT with unit quotas is also NP-hard [27], so if we require both lower and upper quotas to be equal to one for all companies then the two problems are equivalent. Furthermore, no mechanism that finds a stable matching whenever there exists one can be strategyproof.

2.2 Adding types and distributional constraints

In our first application, the organisers want to distribute the foreign students across the projects almost equally. In our second application, there are target numbers for the total number of Hungarian, European and other participants and there are also specific lower quotas for Hungarian students by some companies. These applications motivate our problems with applicant types and distributional constraints.

Let $\mathcal{T} = \{T^1, \ldots, T^p\}$ be the set of types, where $t(a_i)$ denotes the type of applicant a_i. For a company c_j, let l_j^k and u_j^k denote the lower and upper quota for the number of assignees of type T^k. Furthermore, we may also set lower and upper quotas for any type of applicants for a set of companies. In particular, we denote the lower and upper quotas for the total number of applicants of type T^k assigned in the matching by L^k and U^k, respectively. The set of feasibility constraints for the matching is now extended with these lower and upper quotas. Yet, the original stability condition, which does not consider the types of the applicants, remains the same.

3 Solution concepts and integer programming formulations

In all of our formulations we use binary variables $x_{ij} \in \{0, 1\}$ for each application coming from applicant a_i to company c_j. This can be seen as a characteristic function of the matching, where $x_{ij} = 1$ corresponds to the case when a_i is assigned to c_j.

When describing the integer formulations, first we keep the stability condition fixed while we implement the set of distributional constraints. Then we investigate the ways one can relax stability or find most-stable solutions under the distributional constraints.

3.1 Finding stable solutions under distributional constraints

In this subsection we gradually add constraints to the model by keeping the classical stability condition.

Classical HR instance

First we describe the basic IP formulation for HR described in [8]. The feasibility of a matching can be ensured with the following two sets of constraints.

$$\sum_{j:(a_i,c_j)\in E} x_{ij} \leq 1 \text{ for each } a_i \in A \tag{1}$$

$$\sum_{i:(a_i,c_j)\in E} x_{ij} \leq u_j \text{ for each } c_j \in C \tag{2}$$

Note that (1) implies that no applicant can be assigned to more than one company, and (2) implies that the upper quotas of the companies are respected.

To enforce the stability of a feasible matching we can use the following constraint.

$$\left(\sum_{k:r_{ik}\leq r_{ij}} x_{ik}\right) \cdot u_j + \sum_{h:(a_h,c_j)\in E, s_{hj} > s_{ij}} x_{hj} \geq u_j \text{ for each } (a_i, c_j) \in E \tag{3}$$

Note that for each $(a_i, c_j) \in E$, if a_i is matched to c_j or to a more preferred company then the first term provides the satisfaction of the inequality. Otherwise, when the first term is zero, then the second

term is greater than or equal to the right hand side if and only if the places at c_j are filled with applicants with higher scores.

Among the stable solutions we can choose the applicant-optimal one by minimising the following objective function.

$$\sum_{(a_i,c_j)\in E} r_{ij} \cdot x_{ij}$$

Modification for HRT

When the companies can express ties the following modified stability constraints, together with the feasibility constraints (1) and (2), lead to stable matchings. Note that here the only difference between this and the previous constraint is that the strict inequality $s_{hj} > s_{ij}$ became weak.

$$\left(\sum_{k:r_{ik}\leq r_{ij}} x_{ik}\right) \cdot u_j + \sum_{h:(a_h,c_j)\in E, s_{hj}\geq s_{ij}} x_{hj} \geq u_j \text{ for each } (a_i, c_j) \in E \tag{4}$$

Extension with lower quotas

Here, we only add the lower quotas for every company.

$$\sum_{i:(a_i,c_j)\in E} x_{ij} \geq l_j \text{ for each } c_j \in C \tag{5}$$

Adding distributional constraints

As additional constraints we require the number of assignees of a particular type to be between the lower and upper quotas for that type at a company.

$$\sum_{i:t(a_i)=T^k,(a_i,c_j)\in E} x_{ij} \leq u_j^k \text{ for each } c_j \in C \text{ and } T^k \in \mathcal{T} \tag{6}$$

$$\sum_{i:t(a_i)=T^k,(a_i,c_j)\in E} x_{ij} \geq l_j^k \text{ for each } c_j \in C \text{ and } T^k \in \mathcal{T} \tag{7}$$

We can also add similar constraints for set of companies, or for the overall number of different assignees at all companies. We describe the latter, as we will use it when solving our second application.

$$\sum_{i,j:t(a_i)=T^k,(a_i,c_j)\in E} x_{ij} \leq U^k \text{ for each } T^k \in \mathcal{T} \tag{8}$$

$$\sum_{i,j:t(a_i)=T^k,(a_i,c_j)\in E} x_{ij} \geq L^k \text{ for each } T^k \in \mathcal{T} \tag{9}$$

3.2 Relaxing stability

Adding additional constraints to the problem can cause the lack of a stable matching, even if we added some flexibility with the ties.

One way to find a most-stable solution is to introduce nonnegative deficiency variables, d_{ij} for each application and add them to the left side of the stability constraint (4). By minimising the sum of these deficiencies as a first objective we can obtain a solution which is close to be stable.

$$\left(\sum_{k:r_{ik}\leq r_{ij}} x_{ik}\right)\cdot u_j + \sum_{h:(a_h,c_j)\in E, s_{hj}\geq s_{ij}} x_{hj} + d_{ij} \geq u_j \text{ for each } (a_i,c_j)\in E \tag{10}$$

Note that here, if a pair (a_i, c_j) is blocking for the assignment then we need to add more compensation d_{ij} if the number of assignees at c_j that the company prefers to a_i is large. This approach can be reasonable if we want to avoid the refusal of a very good candidate at a company. We call this solution as *matching with minimum deficiency*.

Alternatively, if we just want to minimise the number of blocking pairs then we can set d_{ij} to be binary and minimise the sum of these variables under the following modified constraints.

$$\left(\sum_{k:r_{ik}\leq r_{ij}} x_{ik}\right)\cdot u_j + \sum_{h:(a_h,c_j)\in E, s_{hj}\geq s_{ij}} x_{hj} + d_{ij}\cdot u_j \geq u_j \text{ for each } (a_i,c_j)\in E \tag{11}$$

Here, every blocking pair should be compensated by the same amount, so the number of blocking pairs in minimised. Note that this concept has already been studied in the literature for various models under the name of *almost stable matchings*, see e.g. [14].

3.3 Adjusting upper capacities, envy-free matchings

A different way of enforcing the lowers quota is to relax stability by artificially decreasing the capacities of the companies. This was also the solution in the resident allocation scheme in Japan [22], where the government introduced artificial upper quotas for each of the hospitals, so that in each region the sum of these artificial upper bounds summed up to the target capacity for that region. In the case of our motivating example of project allocation, one simple way of achieving the lower quotas was by reducing the upper quotas at every company.

In this solution what we essentially get is a so-called *envy-free matching*, studied in [33]. The matching is stable with respect to the artificial upper quotas, which means that the only blocking pairs that may occur with regard to the original upper quotas are due to the empty slots created by the difference between the original and the artificial quotas.

However, one may not want to reduce the upper quotas of the companies in the same way, perhaps some more popular companies should be allowed to have more students than the less popular ones. Furthermore, maybe the decision on which upper quotas should be reduced should be make depending on their effect of satisfying the lower quotas (or other requirements). Thus, we may not want to set the artificial upper quotas in advance, but keep them as variables, by ensuring envy-freeness in a different way. One alternative way of enforcing envy-freeness is by the following set of constraints.

$$\sum_{k:r_{ik}\leq r_{ij}} x_{ik} \geq x_{hj} \quad \forall (a_i,c_j),(a_h,c_j)\in E, s_{ij} > s_{hj} \tag{12}$$

Constraints (12) will ensure envy-freeness, by making sure that if applicant a_h is assigned to company c_j and applicant a_i has higher score than a_h at c_j then a_i must be assigned to c_j or to a more preferred company.

3.4 Type-specific priorities

So far we have only considered different approaches of relaxing stability or enlarging the set of feasible solutions in order to satisfy the distributional constraints. In this subsection we study alternative solution concepts and methods for the case when the distributional constraints are type-dependent. This is the case also in our motivating application, where special requirements are set for the foreign students assigned to the companies.

When the number of students of a type does not achieve the minimum required at a place then there are two well-known approaches. For instance in a school choice scenario, where the ratio of an socio-ethnic

group should be improved (see e.g. [2]) then one possible affirmative action is to increase the scores of that group of students as much as needed. The other usual solution is to set some reserved seats to those students (see e.g. [6]).

In our project allocation application our requirement is to have at least one foreign student assigned to every company. If in a stable solution this condition would be violated for a company then we can try to enforce the admission of a foreign student by increasing the scores of the foreign students at this company. We call such a solution as *stable matching with type-specific scores*, where the classical stability condition is required for the adjusted scores. The second approach is to devote one place at each company to foreign students. For this one seat the foreign students will have higher priority than the locals irrespective of their scores, but for the rest of the spaces the usual score-based rankings apply. We call this concept as *stable matching with reserved seats for types*. Note that neither of these two concepts can always ensure that we get at least one foreign student at each company, since they may all have high scores and they may all dislike a particular company. However, this situation changes if we also allow to decrease the scores of a group of students. We will describe this case after discussing the third approach.

Finally, as a third approach, we can also extend the concept of envy-free matchings for types. We do not require any stability with regard to students of different types, but we do require envy-freeness for students of the same type. Thus the so-called *type-specific envy-free matchings* will be those who satisfy the following set of constraints.

$$\sum_{k:r_{ik}\leq r_{ij}} x_{ik} \geq x_{hj} \quad \forall (a_i,c_j),(a_h,c_j)\in E, s_{ij}>s_{hj}, t(a_i)=t(a_h)=T^k \text{ for each } T^k\in\mathcal{T} \qquad (13)$$

That is, if a_i and a_h have the same type and a_h is assigned to c_j then the higher ranked a_i must also be assigned to c_j or to a more preferred company. Note that with this modification we extend the set of feasible solutions compared to the set of envy-free matchings. Another important observation that is motivated by our project allocation problem is that under some realistic assumptions a type-specific envy-free matching always exists, that we will show in the following theorem.

Theorem 1 *Suppose that all the companies are acceptable to every student and that the sum of the lower quotas with regard to each type is less than equal to the number of students of that type, and the sum of the lower quotas across types for a company is less than or equal to the upper quota of that company, then a complete within-type envy-free matching always exists and can be found efficiently.*

PROOF: We construct a within-type envy-free matching separately for each type and then we merge them. When considering a particular type T^k, we set artificial upper quotas at the companies to be equal to the type-specific lower quotas (i.e. l_j^k for company c_j) and we find a stable matching M_k for this type. This stable matching must exist, since we assumed that all the companies are acceptable to every student and the number of students in every type is at least as much as the sum of the lower quotas for that type. We create matching M by merging the stable matchings for the types, i.e. $M = M_1 \cup M_2 \cup \cdots \cup M_p$. Note that no upper quota is violated in M, since we assumed that the sum of the lower quotas across types for any company c_j is less than equal to the upper quota of c_j. By the stability of M_k for every type T^k it follows that matching M is within-type envy-free. If there is still a company c_j, where the overall lower quota (l_j) is not yet met, then we increase an artificial upper quota for some at c_j so that there is still some unmatched applicants of this type. Since the total number of applicants is greater or equal to the sum of the lower quotas, we have to achieve the lower quotas at all companies in this way. Finally, it there are still some unmatched applicants then we increase some artificial upper quotas for their types one-by-one, by making sure that we never exceed any overall upper quota. At the end of this iterative process we must reach a complete within-type envy free matching. □

Let us abbreviate a *complete within-type envy-free matching* as CWTEFM. Now, we will compare this concept of CWTEFM with stable matchings with type-specific scores and observe that they are essentially the same.

Theorem 2 *Under the assumptions of Theorem 1 a complete matching is within-type envy-free if and only if it is stable with type-specific scores.*

PROOF: Suppose first that M is a complete stable matching with type-specific scores, we will see that M is also within-type envy-free by definition. Suppose for a contradiction that there is a student a_i who has justified envy against student a_h of the same type at company c_j, i.e. a_h is assigned to c_j whilst a_i has higher score at c_j than a_h and a_i is assigned to a less preferred company. This would mean that the pair $\{a_i, c_j\}$ is blocking for the adjusted scores, since both students get the same adjustment at c_j, contradicting with the stability of M.

Suppose now that M is a CWTEFM. Let us adjust the scores of the students according to their types at each company such that the weakest students admitted have the same scores across types. Matching M is stable with regard to the adjusted scores, because if a student a_i is not admitted to a company c_j and any better place of her preference that it must be the case that her score at c_j was less then or equal to the score of the weakest assigned student of the same type at c_j, which means that the adjusted score of a_i at c_j is less than or equal to the adjusted score of every assigned student at c_j. □

Instead of using the above described processes of setting type-specific artificial upper quotas or making adjustments for the scores of different types, we can also get a CWTEFM directly by an IP formulation. We shall simply use the feasibility and distributional constraints together with (13) and with an objective function maximising the number of students assigned. This approach is not just more robust than the above described two heuristics, but it has also the advantage that we can enforce additional optimality or fairness criteria. Regarding optimality, we may want to minimise the total rank of the students, leading to a Pareto-optimal assignment under the constraints. As an additional fairness criterion we may aim to minimise the envy across types. We can achieve this by adding deficiency variables to the left hand side of constraints (12) for students of different types, as described in (14) below, and then minimising the sum of the deficiencies. We refer to this solution as MinDefCWTEFM, that is *complete within-type envy-free matching with minimum deficiency across types.*

$$\sum_{k:r_{ik}\leq r_{ij}} x_{ik} + d^j_{ih} \geq x_{hj} \quad \forall (a_i, c_j), (a_h, c_j) \in E, s_{ij} > s_{hj}, t(a_i) \neq t(a_h) \tag{14}$$

We remark that in our project allocation application the conditions of Theorem 1 are satisfied, since all the students have to rank (and accept) all the companies and we require to have at least one foreign student at each company, where the number of foreign students is more than the number of companies. Therefore a complete within-type envy-free matching always exists. Within this set of solutions we decided to minimise the envy across types, as suggested above as first objective. As a secondary objective we can choose to minimise the total rank of the students or as an alternative we can also minimise the open-slot blockings (i.e. the blockings due to unfilled positions with regard to the original upper quotas). The latter objective is useful to make sure that the popular companies always fill their upper quotas, and so the less popular companies will admit fewer students.

4 Further notes

We applied the above described solution concepts for our two motivating applications, in a project allocation problem at Corvinus University of Budapest and for a conference organisation case, that we will describe in details in an extension of this paper. We will also test these solution concepts on randomly generated instances to find out how large problems can be solved with the IP technique.

One could also try to come up with alternative solution concepts and different IP formulations for the same concepts that we proposed. Finally, it would be interesting to test the applicability of our approach in other applications, such as the resident allocation problem with regional quotas, controlled school choice, and college admissions with affirmative action or minority reserves.

References

[1] A. Abdulkadiroglu. College admissions with affirmative action. *International Journal of Game Theory*, 33(4):535–549, 2005.

[2] A. Abdulkadiroglu and L. Ehlers. Controlled School Choice. *Working paper*, 2007.

[3] A. Abdulkadiroğlu, P.A. Pathak, and A.E. Roth. The New York City high school match. *American Economic Review, Papers and Proceedings*, 95(2):364–367, 2005.

[4] A. Abdulkadiroğlu, P.A. Pathak, A.E. Roth, and T. Sönmez. The Boston public school match. *American Economic Review, Papers and Proceedings*, 95(2):368–371, 2005.

[5] K.Cs. Ágoston, P. Biró and I. McBride. Integer programming methods for special college admissions problems. *Journal of Combinatorial Optimization*, 32(4):1371–1399, 2016.

[6] O. Aygün and I. Bo. College Admission with Multidimensional Reserves: The Brazilian Affirmative Action Case. *Working paper*, 2013.

[7] O. Aygün and B. Turhan. Dynamic reserves in matching markets: Theory and applications. *Working paper*, 2016.

[8] M. Baïou and M. Balinski. The stable admissions polytope. *Mathematical Programming*, 87(3), Ser. A:427–439, 2000.

[9] P. Biró. Matching Practices for Secondary Schools – Hungary. `matching-in-practice.eu`, accessed on 23 August 2014

[10] P. Biró. University admission practices - Hungary. `matching-in-practice.eu`, accessed on 23 August 2014

[11] P. Biró, R.W. Irving and I. Schlotter. Stable matching with couples – an empirical study. *ACM Journal of Experimental Algorithmics*, 16, Article No.: 1.2, 2011.

[12] I. Bo. Fair implementation of diversity in school choice. *Games and Economic Behavior*, 97:54–63, 2016.

[13] P. Biró, I. McBride and D.F. Manlove. The Hospitals / Residents problem with Couples: Complexity and Integer Programming models. *In Proceedings of SEA 2014: the 13th International Symposium on Experimental Algorithms*, vol 8504 of LNCS, pp 10-21, Springer, 2014.

[14] I. McBride, D.F. Manlove, and J. Trimble. "Almost stable" matchings in the Hospitals / Residents problem with Couples. *Constraints*, 22(1):50–72, 2016.

[15] J. Drummond, A. Perrault, and F. Bacchus. SAT is an effective and complete method for solving stable matching problems with couples. *In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-15)*, 2015.

[16] F. Echenique and M.B. Yenmez. How to control controlled school choice. *American Economic Review*, 105(8):2679–2694, 2015.

[17] L. Ehlers, I.E. Hafalir, M.B. Yenmez and M.A. Yildirim. School choice with controlled choice constraints: Hard bounds versus soft bounds. *Journal of Economic Theory*, 153:648–683, 2014.

[18] T. Fleiner and N. Kamiyama. A Matroid Approach to Stable Matchings with Lower Quotas. *Mathematics of Operations Research*, 41(2):734–744, 2016.

[19] D. Gale and L.S. Shapley. College Admissions and the Stability of Marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.

[20] M. Goto, F. Kojima, R. Kurata, A. Tamura and M. Yokoo. Designing Matching Mechanisms Under General Distributional Constraints. *American Economic Journal: Microeconomics*, forthcoming, 2017.

[21] R.W. Irving and D.F. Manlove. Approximation algorithms for hard variants of the stable marriage and hospitals/residents problems. *Journal of Combinatorial Optimization*, 16:279–292, 2008.

[22] Y. Kamada, and F. Kojima. Efficient Matching Under Distributional Constraints: Theory and Applications. *American Economic Review*, 105(1):67–99, 2014.

[23] Y. Kamada, and F. Kojima. Stability concepts in matching under distributional constraints. *Journal of Economic Theory*, 168:107–142, 2017.

[24] F. Kojima. School choice: Impossibilities for affirmative action. *Games and Economic Behavior*, 75(2):685–693, 2012.

[25] A. Kwanashie and D.F. Manlove. An Integer Programming approach to the Hospitals / Residents problem with Ties. *In Proceedings of OR 2013: the International Conference on Operations Research*, pages 263-269, Springer, 2014.

[26] D.F. Manlove Algorithms of Matching Under Preferences. *World Scientific Publishing*, 2013.

[27] D.F. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279, 2002.

[28] E. Ronn. NP-complete stable matching problems. *Journal of Algorithms*, 11:285–304, 1990.

[29] A.E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 6(4):991–1016, 1984.

[30] A.E. Roth. On the allocation of residents to rural hospitals: a general property of two-sided matching markets. *Ecnometrica*, 54(2):425–427, 1986.

[31] A.E. Roth and E. Peranson The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design. *American Economic Review*, 89:748–780, 1999.

[32] A.E. Roth and M.A.O. Sotomayor. Two-sided matching: a study in game-theoretic modeling and analysis. *Cambridge: Econometric Society monographs*, 1990.

[33] Wu, Q. and Roth, A.E. (2016). The lattice of envy-free matchings. mimeo

Tverberg plus minus

Imre Bárány

Alfréd Rényi Institute of Mathematics,
Hungarian Academy of Sciences
H-1364 Budapest Pf. 127 Hungary
and
Department of Mathematics
University College London
Gower Street, London, WC1E 6BT, UK
barany.imre@renyi.mta.hu

Abstract: We prove a Tverberg type theorem: Given a set $A \subset \mathbb{R}^d$ in general position with $|A| = (r-1)(d+1)+1$ and $k \in \{0, 1, \ldots, r-1\}$, there is a partition of A into r sets $A_1, \ldots, A_r$ (where $|A_p| \leq d+1$ for each p) with the following property. The unique $z \in \bigcap_{p=1}^r \text{aff}\, A_p$ can be written as an affine combination of the elements in A_p: $z = \sum_{x \in A_p} \alpha(x)x$ for every p and exactly k of the coefficients $\alpha(x)$ are negative. The case $k = 0$ is Tverberg's classical theorem.

Keywords: Tverberg's theorem, sign conditions

1 Introduction and main result

Assume $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$ where $n = (r-1)(d+1)+1$ and $r \geq 2$, $d \geq 1$ are integers. Suppose further that the coordinates of the a_i (altogether dn real numbers) are algebraically independent. A partition $\mathcal{A} = \{A_1, \ldots, A_r\}$ of A is called *proper* if $1 \leq |A_p| \leq d+1$ for every $p \in [r]$. Here and in what follows $[r]$ stands for the set $\{1, \ldots, r\}$. We will show later (Proposition 3) that in this case the intersection of the affine hulls of the A_p is a single point z, that is, $z = \bigcap_{p=1}^r \text{aff}\, A_p$. Equivalently, the following system of linear equations has a unique solution:

$$z = \sum_{x \in A_p} \alpha(x)x \text{ and } 1 = \sum_{x \in A_p} \alpha(x) \text{ for all } p \in [r]. \tag{1.1}$$

One form of Tverberg's classical theorem puts extra conditions on the coefficients $\alpha(x)$.

Theorem 1 *Under the above conditions there is a partition of A into sets $A_1, \ldots, A_r$ such that all $\alpha(x) \geq 0$. In other words, $z = \bigcap_{p=1}^r \text{conv}A_p$.*

This means that the unique solution to (1.1) has $\alpha(x) > 0$ for all $x \in A$. Can we require here that exactly one (or two or more) of the $\alpha(x)$ are negative? A partial answer comes from the next theorem which is the main result of this paper.

Theorem 2 *Assume $k \in \{0, 1, \ldots, r-1\}$. Under the above conditions there is a (proper) partition of A into r parts so that in the unique solution to (1.1) $\alpha(x) < 0$ for exactly k elements $x \in A$.*

Of course the same holds for any set A of n points in $\mathbb{R}^d$, we only have to relax the condition $\alpha(x) < 0$ to $\alpha(x) \leq 0$ for k elements $x \in A$ and $\alpha(x) \geq 0$ for the rest.

It is not clear for what other values of $k \in [n]$ the theorem holds. Certainly $k \leq n - r$ as every A_p contains an x with $\alpha(x) > 0$.

N_1				$-I_d$
1 1 ... 1				
	N_2			$-I_d$
	1 1 ... 1			
		$\ddots$		
			N_r	$-I_d$
			1 1 ... 1	

Table 1: The matrix M, the empty regions indicate zeros

The case of $r = 2$, that is, Radon (plus minus) partitions can be checked directly. Then $|A| = d + 2$ and the outcome is that for any $k \in \{0, 1, \ldots, \lfloor \frac{d+2}{2} \rfloor\}$ there is a partition with exactly k negative $\alpha(x)$. Further, there are examples showing that this does not hold for $k > \lfloor \frac{d+2}{2} \rfloor$. This case is easy as everything is governed by the unique affine dependence of the vectors in A. We omit the details.

The case $d = 1$ is very simple. Then $n = 2r - 1$ and there is no r-partition with r or more negative coefficients, so the trivial bound $k \leq n - r = r - 1$ is tight. In the case $d = 2,\ r = 3$ and $n = 7$ Theorem 2 gives a suitable partition for $k = 0, 1, 2$. A careful case analysis shows that the statement holds for $k = 3$ as well, and an extensive computer aided search did not find any example where it fails to hold for $k = 4$.

We mention further that the proof works for $k = 0$. So it is a new proof of Tverberg's theorem. In fact, the method is the same 'moving the points' (which is sometimes called the 'variational method') but the technique is different. Here is the fact that we need about the intersection of the affine hulls.

Proposition 3 *Assume $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$ with the coordinates of the a_i are algebraically independent and $r \geq 2$, $d \geq 1$ are integers. If the partition $\mathcal{A} = \{A_1, \ldots, A_r\}$ of A is proper and $n = (r-1)(d+1)+1$, then $\bigcap_{p=1}^{r} \mathrm{aff}\, A_p$ is a single point. If $n \leq (r-1)(d+1)$, then $\bigcap_{p=1}^{r} \mathrm{aff}\, A_p = \emptyset$*

The first part must be known, see for instance [2] or [1] for similar statements. The second part is proved in [3]. We give a simple proof in the last section.

2 Preparations and sketch of proof

We write the equation (1.1) in matrix form $M\alpha = b$. The $(n + d) \times (n + d)$ matrix M is made up of blocks. The block corresponding to A_p is a $(d+1) \times |A_p|$ matrix N_p whose columns are the vectors in A_p appended with a 1 in the last row. There are r further blocks, each one is $-I_d$, the negative $d \times d$ identity matrix. They are in the last d columns of M, with a row of zeroes between them. These submatrices are arranged in M as shown on Table 1. All other entries of M are zeroes. The ith column of M corresponds to the vector a_i. Note that $M = M(\mathcal{A})$ depends on A and on the partition $\mathcal{A} = \{A_1, \ldots, A_r\}$ as well.

The variables are $\alpha = (\alpha_1, \ldots, \alpha_n, z_1, \ldots, z_d)^T \in \mathbb{R}^{n+d}$ and the right hand side vector is $b \in \mathbb{R}^{n+d}$ that has coordinate zero everywhere except in positions $d+1, 2(d+1), \ldots, r(d+1)$ where it has one. The original system (1.1) is the same as

$$M\alpha = b. \tag{2.1}$$

Let M_i denote the matrix obtained by replacing the ith column of M by the vector b. We will need the following fact.

Proposition 4 *If the partition of A is proper, then* $\det M \neq 0$ *and* $\det M_i \neq 0$ *for all* $i \in [n]$.

PROOF: The system (1.1) or, what is the same, (2.1) has a unique solution iff $\det M \neq 0$, which happens iff $\bigcap_{p=1}^{r} \operatorname{aff} A_p$ is a single point. So $\det M \neq 0$ is implied by Proposition 3. Next $\det M_i = 0$ implies by Cramer's rule that $\alpha_i = \frac{\det M_i}{\det M} = 0$ and so $\bigcap_{p=1}^{r} \operatorname{aff}(A_p \setminus \{a_i\}) \neq \emptyset$ which is impossible according to second half of the same Proposition. □

As in Tverberg's original proof [3], we use the method of moving the points. As a first step we find an initial set $A_0 \subset \mathbb{R}^d$ that has property [k], that is, A_0 has a (proper) partition where the unique solution to (2.1) has exactly k negative αs. This is easy. Start with a single point a_1 contained in the interior of $r-1$ d-dimensional simplices. Then translate the first k simplices so that a_1 ends up on the other side of exactly one of the hyperplanes defining the simplex. Then a_1 and the vertices of the simplices form the required set A_0. It is also clear that this set A_0 can be chosen so that its points are in algebraically independent position.

For moving the points we begin with $n+1$ vectors $b_1, a_1, \ldots, a_n \in \mathbb{R}^d$ that have algebraically independent coordinates. We assume that $A = \{a_1, \ldots, a_n\}$ has property [k] and are going to show that so does $\{b_1, a_2, \ldots, a_n\}$. This is the main step of the proof. Once it is done, the proof is complete: we repeat the main step n times to arrive from the intitial set A_0 to the target set.

For the main step we define $a(t) = (1-t)a_1 + tb_1$ and check that the set $A(t) = \{a(t), a_2, \ldots, a_n\}$ has property [k] for all $t \in [0,1]$ except possibly for finitely many values of t (to be called *critical values*). $A(t)$ has property [k] iff it has a partition $\mathcal{A}(t) = \{A_1(t), A_2, \ldots, A_r\}$ such that in the unique solution to (2.1) exactly k of the α_i are negative. What we are going to show, actually, is that the partition $\mathcal{A}(t)$ remains the same between two consecutive critical values. So $\mathcal{A}(t)$ only changes at the critical values.

We will always consider $a(t)$ as the first vector in $A_1(t)$. Sometimes we write A_1 instead of $A_1(t)$ suppressing the dependence on t.

Clearly $\bigcap_{p=1}^{r} \operatorname{aff} A_p \neq \emptyset$ happens iff the system

$$M(t)\alpha = b \tag{2.2}$$

has a solution. Here $M(t)$ is the same matrix as M with the same partition $\mathcal{A}$ except that in the first column a_1 is replaced by $a(t)$. For $i \in [n]$ we write $M_i(t)$ when ith column of $M(t)$ is replaced by b.

Proposition 5 *The equation* $\det M(t) = 0$ *has at most one solution. The same holds for equation* $\det M_i(t) = 0$.

PROOF: Note that $\det M(t)$ is a linear function of t so it either has a single root or it is constant. If it is a constant, then $\det M(t) = \det M(0)$ which is non-zero according to Proposition 4. The same applies to $\det M_i(t)$. □

Assume that, for some $t_0 \in [0,1]$, $A(t_0)$ has property [k] with some fixed (and proper) partition $\mathcal{A}(t) = \{A_1(t), \ldots, A_r\}$ and $\det M(t_0) \neq 0$. This is the case for $t_0 = 0$, actually. So the unique solution to $M(t)\alpha = b$ has exactly k negative and $n-k$ positive α_is. Then this holds in a small neighbourhood of t_0, let $\tau > t_0$ be the smallest number where this condition fails. So at the critical value τ, either $\det M(\tau) = 0$ or $\alpha_j(\tau) = 0$ for some $j \in [n]$. Note that $a_j \neq a(t)$ (or $j \neq 1$) here as, by Cramer's rule, $\alpha_1(t) = \frac{\det M_1(t)}{\det M(t)}$ and $\det M_1(t)$ is a non-zero constant (independent of t).

The key step in the proof of Theorem 2 is the next lemma.

Lemma 6 *Under the above conditions there is* $\varepsilon > 0$ *such that for* $t \in (\tau, \tau+\varepsilon)$ $A(t)$ *has property [k] with another partition* $\mathcal{A}'(t)$.

With the lemma, the proof of Theorem 2 is finished as follows. Start with $t_0 = 0$ where $A(t_0)$ has property [k] with partition $\mathcal{A}(t_0)$. With this partition $\det M(t_0) \neq 0$. Let $\tau_0 > t_0$ be the next critical

value. By the lemma, for a suitable small $\varepsilon > 0$, $A(t)$ has property [k] for all $t \in (\tau_0, \tau_0 + \varepsilon)$ with a proper partition $\mathcal{A}_1(t) = \mathcal{A}'(t)$. Choose some $t_1 \in (\tau_0, \tau_0 + \varepsilon)$ with $\det M(t_1) \neq 0$ and let $\tau_1 > t_1$ be the next critical value. Apply the lemma again. It gives some $\varepsilon > 0$ such that $A(t)$ has property [k] for all $t \in (\tau_1, \tau_1 + \varepsilon)$ with a proper partition $\mathcal{A}_2(t)$, and so on. In this induction argument the intervals (τ_k, τ_{k+1}) cover every $t \geq 0$ (except the critical values) because Proposition 5 implies that there are only finitely many critical values. So $t = 1$ is also covered. $\square$

3 Proof of Lemma 6

We begin with the following claim.

Claim 7 *At the critical value* τ $\det M(\tau) \neq 0$.

Proof: Assume that, on the contrary, $\det M(\tau) = 0$. For $t \in [t_0, \tau)$ $\det M(t) \neq 0$ and we may assume it is positive (by swapping two columns in the same part of the partition if necessary). Since $k < r$ there is a part A_p with $\alpha_i(t) > 0$ for all i with $a_i \in A_p$ ($p = 1$ is possible). The condition $\sum_{i:a_i \in A_p} \alpha_i(t) = 1$ is the same, by Cramer's rule, as

$$\sum_{i:a_i \in A_p} \det M_i(t) = \det M(t).$$

All terms here are positive and $\det M(t) \to 0$ as $t \to \tau$. Then $\det M_i(t) \to 0$ as $t \to \tau$ as well. It follows that $\lim_{t\to\tau} \alpha_i(t)$ exists and equals $\alpha_i(\tau) \geq 0$, as both $\det M(t)$ and $\det M_i(t)$ are linear functions of t. Also, $\sum_{i:a_i \in A_p} \alpha_i(\tau) = 1$, and the point $z(\tau) = \sum_{i:a_i \in A_p} \alpha_i(\tau) a_i$ lies in $\mathrm{conv} A_p$. Further, $z(t) = \sum_{j:a_j \in A_q} \alpha_j(t) a_j$ for every other A_q. As $\lim_{t\to\tau} z(t) = z(\tau)$ exists, taking limit here shows that $\lim_{t\to\tau} \alpha_j(t)$ exists and is finite for every j with $a_j \in A_q$. In particular, for $j = 1$ this means that

$$\lim_{t\to\tau} \alpha_1(t) = \lim_{t\to\tau} \frac{\det M_1(t)}{\det M(t)}$$

exists and is finite, so $\det M_1(t) \to 0$. But $\det M_1(t)$ is a non-zero constant, a contradiction. $\square$

Remark. This is the only place in the proof where we use the condition $k < r$. We mention further that $\det M(t) \neq 0$ holds when $z(t) \in \mathrm{conv} A_p$ for some $A_p \in \mathcal{A}(t)$.

Claim 8 *At the critical value* τ *the vector* a_j *with* $\alpha_j(\tau) = 0$ *is unique.*

Proof: As $\det M(\tau) \neq 0$ the limit of $z(t)$ and of each $\alpha_i(t)$ as $t \to \tau$ exists and equals $z(\tau)$ and $\alpha_i(\tau)$ so they are the same $z(\tau)$ and $\alpha_i(\tau)$ as in the proof of the previous claim. They are also the solutions to equation $M(\tau)\alpha = b$.

Assume that $\alpha_j(\tau) = 0$ and $\alpha_i(\tau) = 0$ for distinct $i, j \in [n]$. Here $i, j \neq 1$ as we have seen. This means that $z(\tau)$ lies in the intersection of the affine hulls of the $A_p \setminus \{a_i, a_j\}$ for all p including $A_1(\tau) \setminus \{a_i, a_j\}$. Set $A^1 = A_1(\tau) \cup \{a_1, b_1\} \setminus \{a(\tau), a_i, a_j\}$. Here of course $a(\tau) = (1 - \tau)a_1 + \tau b_1$. Then

$$\mathrm{aff}\,(A_1(\tau) \setminus \{a_i, a_j\}) \subset \mathrm{aff}\, A^1,$$

and so $z(\tau) \in \mathrm{aff}\, A^1 \cap \bigcap_{p=2}^{r} \mathrm{aff}\,(A_p \setminus \{a_i, a_j\})$. But the union of the sets $A^1, A_2, \ldots, A_r$ only contains $n - 1$ points. So by Proposition 3, their affine hulls have no point in common. $\square$

Thus the unique a_j belongs to a unique A_q, $q = 1$ is possible. Note that $|A_q| > 1$ as otherwise there is only $a_j \in A_q$ and $\alpha_j(\tau) = 0$ so the sum $\sum_{i:a_i \in A_q} \alpha_i(\tau) = 0$ while it should be 1.

We assume again that $\det M(t) > 0$.

Let $M^*(t)$ be the matrix obtained from $M(t)$ by replacing its jth column by $(a_j, 1, a_j, 1, \ldots, a_j, 1)^T \in \mathbb{R}^{n+d}$ (r copies of $(a_j, 1) \in \mathbb{R}^{d+1}$). The linear system $M^*(\tau)\alpha = b$ has two solutions, namely, the solution

to $M(\tau)\alpha = b$ (since $\alpha_j(\tau) = 0$) and the vector $(\beta_1, \ldots, \beta_n, v_1, \ldots, v_d)^T \in \mathbb{R}^{n+d}$ where $\beta_i = 1$ if $i = j$ and $\beta_i = 0$ otherwise, and $(v_1, \ldots, v_d) = a_j$. This implies that $\det M^*(\tau) = 0$.

Next let $M^p(t)$ be the matrix obtained from $M^*(t)$ by replacing its jth column by $(0, \ldots, 0, a_j, 1, 0, \ldots, 0)^T \in \mathbb{R}^{n+d}$; here the first $(p-1)(d+1)$ and the last $(r-p)(d+1)$ entries are zero, and between these zeroes sits $(a_j, 1) \in \mathbb{R}^{d+1}$. Expand $\det M^*(t)$ along its jth column:

$$\det M^*(t) = \sum_1^r \det M^p(t).$$

Define $Q = \{q\} \bigcup \{p \in [r] : |A_p| \leq d\}$. Recall that $a_j \in A_q$. It is evident that $\det M^p(t) = 0$ if $p \notin Q$ as the jth column and the ith columns with $a_i \in A_p$ are linearly dependent. Thus

$$\det M^*(t) = \sum_{p \in Q} \det M^p(t).$$

Observe now that $M(t) = M^q(t)$ and so $\det M^q(t)$ is positive at $t = \tau$. As the left hand side is zero at $t = \tau$ there must be an $s \in Q$ with $\det M^s(\tau) < 0$. Define now a new partition $\mathcal{A}'(t) = \{A'_1(t), A'_2, \ldots, A'_r\}$ of $A(t)$ as follows: $A'_q = A_q \setminus \{a_j\}$, $A'_s = A_s \cup \{a_j\}$, and $A'_p = A_p$ in all other cases. Observe that $M^s(t)$ is the matrix corresponding to the new partition $\mathcal{A}'(t)$. Note that this partition is proper because $s \in Q$.

We are almost finished now. We **claim** that the new partition satisfies the requirements for all t in $(\tau, \tau + \varepsilon)$ for some small $\varepsilon > 0$: exactly k of the $\alpha'_i(t)$ are negative and the rest are positive. First, for all $i \in [n]$

$$\alpha_i(t) = \frac{\det M_i(t)}{\det M(t)} \text{ and } \alpha'_i(t) = \frac{\det M^s_i(t)}{\det M^s(t)}$$

and $\alpha_j(\tau) = \alpha'_j(\tau) = 0$ because $\alpha_j(\tau) = 0$ and $M_j(\tau) = M^s_j(\tau)$. This and the uniqueness of the $\alpha_i(\tau)$ imply that $\alpha_i(\tau) = \alpha'_i(\tau)$ for all $i \in [n]$. Here $\alpha'_i(t)$ depends continuously on t so it has the same (non-zero) sign as $\alpha_i(\tau)$ in a small neighbourhood of τ for all $i \neq j$. When $i = j$, $M^s_j(t) = M_j(t)$ and so

$$\alpha'_j(t) = \frac{\det M^s_j(t)}{\det M^s(t)} = \frac{\det M_j(t)}{\det M^s(t)},$$

while

$$\alpha_j(t) = \frac{\det M_j(t)}{\det M(t)}.$$

Here $\det M_j(t)$ changes sign at $t = \tau$ and the signs of $\det M(t)$ and $\det M^s(t)$ are opposite. Consequently the sign of $\alpha_j(t)$ for $t < \tau$ coincides with that of $\alpha'_j(t)$ for $t > \tau$ and close enough to τ. □

4 A stronger version of Theorem 2

The proof shows that the sign of $\alpha_j(t)$ does not change at the critical value τ. This means that while moving the points, the sign of any $\alpha_i((t)$ remains unchanged, except possibly at a critical value where it may become zero. But even then, the sign of $\alpha_i(t)$ is the same for $t < \tau$ and for $t > \tau$. This gives a strengthening of Theorem 2: one can prescribe in advance which $x \in A$ is going to be negative and which one positive.

Theorem 9 *Assume that, under the conditions of Theorem 2, a set $B \subset A$ is given with $|B| = k$. Then there is a (proper) partition of A into r parts so that the unique solution to (1.1) satisfies $\alpha(x) < 0$ for $x \in B$ and $\alpha(x) > 0$ for $x \in A \setminus B$.*

5 Proof of Proposition 3

As we have seen, $\bigcap_1^r \operatorname{aff} A_p$ is a single point iff the linear system (1.1) or what is the same (2.1) has a unique solution which happens iff $\det M \neq 0$. Here $\det M$ is a polynomial with integral coefficients in the coordinates of the a_i. If this polynomial is zero at some algebraically independent points $a_1, \ldots, a_n$, then it is identically zero. So it suffices to show one example where it is non-zero or, what is the same, one example where $\bigcap_1^r \operatorname{aff} A_p$ is a single point.

This is quite easy. Suppose $|A_p| = d + 1 - m_p$ for all $p \in [r]$ and $m_1 \geq m_2 \geq \ldots \geq m_p$. As $\mathcal{A}$ is a proper partition, $0 \leq m_p \leq d$. Let H_p be the subspace of $\mathbb{R}^d$ defined by equations $x_i = 0$ for $i = \sum_1^{p-1} m_j + 1, \ldots, \sum_1^p m_j$. Since $n = (r-1)(d+1) + 1$, $\sum_1^r m_p = d$, implying that $\bigcap_1^r H_p$ is a single point, namely the origin. For each $p \in [r]$ choose $|A_p|$ affinely independent points in H_p. Their affine hull is exactly H_p, finishing the proof of the first part.

For the second part we can assume that A_p is nonempty for all p, and also that $|A_p| \leq d + 1$ as otherwise one can delete some elements of A_p while keeping its affine hull the same. We suppose further that $n = (r-1)(d+1)$ by adding extra (and algebraically independent) points to some suitable A_ps. Then $\bigcap_1^r \operatorname{aff} A_p \neq \emptyset$ iff the corresponding linear system (2.1) has a solution. Now M is an $(n+1) \times n$ matrix. Adding b to M as a last column we get the matrix M^*. The system (2.1) has a solution iff $\operatorname{rank} M = \operatorname{rank} M^*$. The previous argument shows that $\operatorname{rank} M = n - 1$ and so we have that, as a polynomial, $\det M^*$ is identically zero. Again it suffices to give a single example where $\bigcap \operatorname{aff} A_p = \emptyset$. We use the same example as before except that this time $\sum_1^r m_p = d + 1$ so we can add the equation $\sum_1^d x_i = 1$ to the ones defining H_1 if $m_1 < d$ and then $\bigcap H_p = \emptyset$, indeed. If $m_1 = d$ then $H_1 = 0$ and $m_2 = 1$ and we define H_2 by the single equation $x_1 + x_2 = 1$, and again $\bigcap H_p = \emptyset$. The sets A_p are constructed the same way as above. □

Acknowledgements. Support from ERC Advanced Research Grant no 267165 (DISCONV) and from Hungarian National Research Grants no K111827 and K116769 is acknowledged. I'm also indebted to Attila Pór and Manfred Scheucher for useful and illuminating discussions, and to an anonymous referee for careful reading and valuable comments.

References

[1] Doignon, J.-P., Valette, G.: Radon partitions and a new notion of independence in affine and projective spaces. *Mathematika*, **24** (1977), 86–96.

[2] Perles, M.A., Sigron, M.: Strong general position. (2014), arXiv:1409.2899

[3] Tverberg, H.: A lower bound for the volumes of strictly convex bodies with many boundary points. *J.London Math. Soc.*, **41** (1966), 123–128.

Directed hypergraphs and Horn minimization

Kristóf Bérczi

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
berkri@cs.elte.hu

Erika R. Bérczi-Kovács

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
koverika@cs.elte.hu

Abstract: A Boolean function given in a conjunctive normal form is Horn if every clause contains at most one positive literal, and it is pure Horn if every clause contains exactly one positive literal. Due to their computational tractability, Horn functions are studied extensively in many areas of computer science and mathematics such as combinatorics, artificial intelligence, database theory, algebra and logic.

The present paper considers the problem of finding minimal representations of pure Horn functions. We give a new proof for a recent min-max result of Boros et al. regarding body-minimal representations. The proof is algorithmic and finds the so called Guigues-Duquenne basis. We also describe a new construction that combines two existing representations into a third one.

Keywords: directed hypergraphs, GD basis, Horn minimization

1 Introduction

As a subclass of Boolean functions, Horn functions play an important role in different areas of mathematics due to their interesting computational properties. The satisfiability problem for this subclass of Boolean functions can be solved in linear time and the equivalence of Horn formulas can be decided in polynomial time [10]. This concept appears as lattices and closure systems in algebra, as implicational systems in artificial intelligence, as directed hypergraphs in graph theory, and is also used for representing knowledge base in propositional expert systems.

Informally, the Horn minimization problem is to find a minimal representation that is equivalent to a given Horn formula. For example, such a representation can be used to reduce the size of the knowledge base in a propositional expert system, thus improving the performance of the system. The size of a formula can be measured in many different ways (see [5]). Unfortunately, it is NP-hard to find an optimal representation for almost all of these measures. There is however an interesting exception, called body-minimal representation, for which polynomial time algorithms were independently discovered [5, 9, 11]. In [7], Boros et al. gave an explanation why this measure is so different from the others in terms of tractability by providing a min-max result on the minimum number of bodies appearing in the representation of a Horn function. Their proof is algorithmic and it actually determines a canonical body-minimal representation called the Guigues-Duquenne basis.

A common aspect of previous algorithms for determining a body-minimal representation is that they are using frameworks different from that of directed hypergraphs, for example, functional dependencies or implication systems. For this reason, the steps of these algorithms are difficult to follow and they do not reveal the structure of body-minimal representations. One motivation of our investigations was to give a better understanding of the min-max result of [7] by using a purely graph theoretical approach.

In contrast to body-minimal representations, edge-minimal representations are not only hard to find but even hard to approximate. Bhattacharya et al. [6] showed that this problem is inapproximable

within a factor $2^{O(\log^{(1-\varepsilon)}(n)}$ assuming $NP \subsetneq DTIME(n^{polylog(n)})$, while Boros and Gruber showed that it is inapproximable within a factor $2^{O(log^{1-o(1)}n)}$ assuming $P \subsetneq NP$, where n denotes the number of variables. However, the existence of an $O(n^c)$ approximation for some $0 < c < 1$ is a rather interesting open problem; such an approximation algorithm would immediately find a wide list of applications. We present a surprising result, which given two pure Horn formulas Φ_1 and Φ_2, constructs a new one Φ such that the bodies and heads of Φ form subsets of the bodies of Φ_1 and the heads of Φ_2, respectively. We hope that this observation may help us in finding a good approximation for the edge-minimal representation.

The rest of the paper is organized as follows. A brief introduction into Horn logic is given in Section 2. We give a new algorithmic proof of the min-max result of Boros et al. in Section 3. In Section 4, we show that the body-minimal representation provided by the algorithm is in fact the GD basis. Finally, we show how a new representation from two given ones can be constructed in Section 5.

2 Preliminaries

2.1 Horn logic

Let V be a set of n variables. Members of V are called **positive** while their negations are called **negative literals**. A **boolean function** is a mapping $f : \{0,1\}^V \to \{0,1\}$. For a subset $Z \subseteq V$ let χ_Z denote the **characteristic vector** of Z, that is, $\chi_Z(v) = 1$ if $v \in Z$ and 0 otherwise. Then Z is called **true** if $f(\chi_Z) = 1$ and **false** otherwise. The **sets of true** and **false sets** of f are denoted by $\mathcal{T}_f$ and $\mathcal{F}_f$, respectively.

It is known that any boolean function can be represented by a **conjunctive normal form** (CNF). A CNF is a conjunction of **clauses**, where a clause is a disjunction of literals. A clause is **Horn** if at most one of its literals is positive, and is **pure Horn** (or **definite Horn**) if it contains exactly one positive literal. Given a representation, the **set of clauses** is denoted by $\mathcal{C}$. A CNF $\Phi = (V, \mathcal{C})$ is **pure Horn** if all of its clauses are pure Horn. Finally, a boolean function h is **pure Horn** if it can be represented by a pure Horn CNF. For a subset $\emptyset \neq B \subseteq V$ and $v \in V \setminus B$ we write $(B \to v)$ to denote the pure Horn clause $C = v \vee \bigvee_{u \in B} \overline{u}$. Here B and v are called the **body** and **head** of the clause, respectively. The **set of bodies** and **set of heads** appearing in a CNF representation Φ are denoted by $\mathcal{B}(\Phi)$ and $\mathcal{H}(\Phi)$, respectively.

It is known that for any pure Horn function h, $\mathcal{T}_h$ is closed under intersection and contains V. Moreover, for any set $\mathcal{T}$ of subsets of V which is closed under intersection and contains V, there exists a pure Horn function h with $\mathcal{T}_h = \mathcal{T}$. Hence there is a one-to-one correspondence between pure Horn functions and sets of subsets of V closed under intersection and containing V.

Given a pure Horn function h, the **forward chaining closure** of a set $Z \subseteq V$ is the unique smallest true set containing Z and is denoted by $F_h(Z)$. If Φ is a pure Horn CNF representation of h then the forward chaining closure can be obtained by the following method. Set $F^0_\Phi(Z) := Z$. In a general step, if $F^i_\Phi(Z)$ is a true set then $F_h(Z) = F^i_\Phi(Z)$. Otherwise take an arbitrary violated implication $(B \to v)$ of Φ and set $F^{i+1}_\Phi := F^i_\Phi(Z) + v$. Note that $(B \to v)$ is violated by $F^i_\Phi(Z)$ if and only if $B \subseteq F^i_\Phi(Z)$ but $v \notin F^i_\Phi(Z)$. It is known that the result of the process depends neither on the particular choice of the representation Φ nor on the order in which violated implications are chosen, but only on the underlying function h.

2.2 Directed hypergraphs

Directed hypergraphs are generalizations of directed graphs and can be defined in several ways [8, 12]. In our investigations we will use the following notation. A **directed hypergraph** is a pair $H = (V, \mathcal{E})$ where V is a set of **nodes** and $\mathcal{E}$ is a set of **hyperedges**. A hyperedge is a pair (B, v) where $\emptyset \neq B \subseteq V$ is the **body** and $v \in V \setminus B$ is the **head** of the hyperedge. The **set of bodies** and **set of heads** appearing in H are denoted by $\mathcal{B}(H)$ and $\mathcal{H}(H)$, respectively. We say that a hyperedge $(B, v) \in \mathcal{E}$ **covers** a set $Z \subseteq V$ if $B \subseteq Z$ and $v \notin Z$. The hypergraph H **covers** a family $\mathcal{P}$ of subsets of V if for each $Z \in \mathcal{P}$

there exists an edge in $\mathcal{E}$ covering Z. A subset $Z \subseteq V$ is called **true** if H does not cover Z and **false** otherwise. The **sets of true** and **false sets** are denoted by $\mathcal{T}_H$ and $\mathcal{F}_H$, respectively.

Given a node $v \in V$, let $H - v$ denote the hypergraph obtained from H by deleting each hyperedge containing v (either as a body node or a head node). We say that a node $v \in V$ is **reachable** from a set $Z \subseteq V$ in H if either $v \in Z$ or there exists a hyperedge (B, v) such that each node in B is reachable from Z in $H - v$. The **set of nodes reachable from** Z in H is denoted by $F_H(Z)$.

2.3 Pure Horn functions and directed hypergraphs

There is a natural one-to-one correspondence between pure Horn CNFs and directed hypergraphs. Namely, a CNF $\Phi = (V, \mathcal{C})$ and a hypergraph $H = (V, \mathcal{E})$ correspond to each other if $(B \to v) \in \mathcal{C}$ if and only if $(B, v) \in \mathcal{E}$. Let h be a pure Horn function, Φ be a pure Horn CNF representing h and H be the corresponding hypergraph. It is easy to see that $\mathcal{T}_h = \mathcal{T}_H$, $\mathcal{F}_h = \mathcal{F}_H$, $\mathcal{B}(\Phi) = \mathcal{B}(H)$, $\mathcal{H}(\Phi) = \mathcal{H}(H)$ and $F_h(Z) = F_H(Z)$ for every $Z \subseteq V$. Hence the problem of finding a body-minimal representation of h is equivalent to finding a hypergraph $H = (V, \mathcal{E})$ with $\mathcal{T}_H = \mathcal{T}_h$ and $|\mathcal{B}(H)|$ being minimal. For a given pure Horn CNF $\Phi = (V, \mathcal{C})$, we will denote the corresponding directed hypergraph by $H_\Phi = (V, \mathcal{E}_\Phi)$.

3 Body-minimal representation

Let h be a pure Horn function. A hyperedge (X, v) is called **valid** if it does not cover a true set in $\mathcal{T}_h$. A true set Y **separates** false sets X_1 and X_2 if $X_1 \cap X_2 \subseteq Y$ and either $Y \subset X_1$ or $Y \subset X_2$. Two sets X_1 and X_2 are called **independent** if they can not be covered by valid hyperedges having the same body. Note that two false sets are independent if and only if either they are separated by a true set or $X_1 \cap X_2 = \emptyset$. Observe that a hypergraph $H = (V, \mathcal{E})$ represents h if and only if it covers $\mathcal{F}$ and only has valid hyperedges.

The next min-max result first appeared in [7] in a slightly different form. We give a new proof here using directed hypergraphs. The advantage of using the hypergraph terminology is that both the statement of the theorem and the main steps of the algorithmic proof are easier to interpret.

Theorem 1 *Let h be an arbitrary pure Horn function. The minimum number of bodies appearing in a hypergraph representation $H = (V, \mathcal{E})$ of h equals the maximum number of pairwise independent false sets.*

Proof: Take an arbitrary representation $H = (V, \mathcal{E})$ of h and a family $\mathcal{I}$ of pairwise independent false sets. For each $X \in \mathcal{I}$, there must be a valid hyperedge in $\mathcal{E}$ that covers X. As no two members of $\mathcal{I}$ can be covered by valid hyperedges having the same body, the number of different bodies appearing in the representation is at least $|\mathcal{I}|$, showing $|\mathcal{B}(H)| \geq |\mathcal{I}|$. By choosing H to be body-minimal and $\mathcal{I}$ to be maximal, we get that the minimum is at least the maximum. Hence, in order to prove equality, it suffices to show a representation $H = (V, \mathcal{E})$ of h and a family $\mathcal{I}$ of pairwise independent false sets such that $|\mathcal{B}(H)| = |\mathcal{I}|$.

Procedure MinMax constructs such a representation. At the beginning, we set $H := (V, \emptyset)$. At a general step of the algorithm, take an inclusionwise minimal false set $X \in \mathcal{F}_h$ not covered by H and let $Y \in \mathcal{T}_h$ be the minimal true set containing X. Note that Y is uniquely determined as $\mathcal{T}_h$ is closed under intersection and $V \in \mathcal{T}_h$. Add (X, v) to $\mathcal{E}$ for each $v \in Y - X$.

We repeat these steps as long as possible. Let $H = (V, \mathcal{E})$ be the resulting hypergraph, let $X_1, \ldots, X_t$ denote the bodies in H in the order they got into H and let Y_i be the unique minimal true set containing X_i for $i = 1, \ldots, k$. Clearly, H covers every false set in $\mathcal{F}_h$ and contains only valid hyperedges. In addition, $X_i \in \mathcal{F}_h$ for $i = 1, \ldots, t$. We claim that these false sets are pairwise independent. Indeed, take two sets, say X_i and X_j with $i < j$. If $X_i \subset X_j$ then Y_i separates them, otherwise one of the hyperedges $\{(X_i, v) : \ v \in Y_i - X_i\}$ would cover X_j, hence X_j could not appear as a body in the representation. Assume now that none of $X_i - X_j$, $X_i \cap X_j$ and $X_j - X_i$ is empty. We claim that $Y = X_i \cap X_j$ is a true set. Assume indirectly that Y is false. Then Y became covered no later than X_i and X_j. However, a

hyperedge covering Y also covers at least one of X_i and X_j, contradicting that both of them are bodies in the final hypergraph. Hence Y is a true set which separates X_i and X_j. Thus we conclude that $X_1, \ldots, X_t$ are independent false sets, finishing the proof. □

Procedure MinMax

Input : A pure Horn function h.
Output: A body-minimal representation $H = (V, \mathcal{E})$ of h.

1 $\mathcal{E} := \emptyset$
2 $H := (V, \mathcal{E})$
3 **while** $\exists$ *false set not covered by* H **do**
4 | Choose an inclusionwise minimal false set X not covered by H.
5 | Let Y be the unique minimal true set containing X.
6 | $\mathcal{E} := \mathcal{E} \cup \{(X, v) : \ v \in Y - X\}$
7 **end**
8 Output $H = (V, \mathcal{E})$.

Now we show that Theorem 1 is equivalent to the min-max result of Boros et al. [7]. Let $\Phi = (V, \mathcal{C})$ be a pure Horn CNF. For a subset $S \subseteq V$, define $\mathcal{E}_S = \{(B \to v) \in \mathcal{C} : \ B \subseteq S, \ v \notin S\}$ and call such a set **essential** if it is non-empty. Two essential sets $\mathcal{E}_{S_1}$ and $\mathcal{E}_{S_2}$ where $S_1 \neq S_2$ are **body-disjoint** if no two nodes $v_1 \in F_\Phi(S_1 \cap S_2) \setminus S_1$ and $v_2 \in F_\Phi(S_1 \cap S_2) \setminus S_2$ exist simultaneously. (In fact both essentiality and body-disjointness are defined slightly differently in [7], but for now we can think of these sets as mentioned above.)

Theorem 2 (Boros, Čepek, Makino) *Let h be an arbitrary pure Horn function. Then the minimum number of bodies appearing in a pure Horn CNF representation of h equals the maximum number of pairwise body-disjoint essential sets.*

It is not difficult to see that $\mathcal{E}_{S_1}$ and $\mathcal{E}_{S_2}$ are body-disjoint essential sets if and only if they are false and either $S_1 \cap S_2 = \emptyset$ or $F_\Phi(S_1 \cap S_2)$ is a true set separating S_1 and S_2. Hence the equivalence of the two theorems follows.

By using the notation of the proof of Theorem 1, we get that the pure Horn CNF

$$\Psi = \bigwedge_{i=1}^{t} \bigwedge_{v \in Y_i \setminus X_i} (X_i \to v) \tag{1}$$

is a body minimal representation of h. Hence the proof immediately suggests a direct algorithm for determining a body-minimal representation of a pure Horn function h given by a pure Horn representation Φ.

Theorem 3 *Let h be a pure Horn function given by a pure Horn CNF representation Φ. Then a body-minimal pure Horn representation Ψ defined by* (1) *can be determined in polynomial time.*

Proof: It suffices to show that Steps 4 and 5 of Procedure MinMax can be performed in polynomial time. Assume that the algorithm constructed a hypergraph $H = (V, \mathcal{E})$ so far. Observe that $F_H(Z) \subseteq F_{H_\Phi}(Z)$ for every $Z \subseteq V$ as we only added hyperedges of form (X_i, v) where $v \in F_h(X_i)$.

In Step 4, we have to find a minimal set X which is covered by H_Φ but uncovered by H. Such a set X surely contains a body $B \in \mathcal{B}(H_\Phi)$. As H does not cover X, necessarily we have $F_H(B) \subseteq X$. On the other hand, $F_H(B)$ is covered by H_Φ unless $F_H(B) = F_{H_\Phi}(B)$. Hence X can be chosen to be a minimal set among the sets $F_H(B)$ for $B \in \mathcal{B}(H_\Phi)$ with $F_H(B) \neq F_{H_\Phi}(B)$.

The unique minimal true set containing a given false set X is just $F_h(X)$ which can be determined by using forward chaining (based on Φ), hence Step 5 can be performed easily. □

Procedure BODYMINIMAL

Input : A pure Horn CNF representation Φ of h.
Output: A body-minimal representation Ψ of h.

1 $\mathcal{E} := \emptyset$
2 $H := (V, \mathcal{E})$
3 $\Psi := \emptyset$
4 **while** $\exists B \in \mathcal{B}(H_\Phi) : \ F_H(B) \neq F_{H_\Phi}(B)$ **do**
5 $\quad X := \text{argmin}\{F_H(B) : \ B \in \mathcal{B}(H_\Phi), \ F_H(B) \neq F_{H_\Phi}(B)\}$
6 $\quad Y := F_{H_\Phi}(B)$
7 $\quad \mathcal{E} := \mathcal{E} \cup \{(X, v) : \ v \in Y - X\}$
8 $\quad \Psi := \Psi \wedge \big(\bigwedge_{v \in Y \setminus X} (X \to v) \big)$
9 **end**
10 Output Ψ.

A short description of the direct algorithm is presented by Procedure BODYMINIMAL.

The proofs of Theorems 1 and 3 imply the following, somewhat surprising result.

Theorem 4 *Let h be a pure Horn function given by a pure Horn CNF representation Φ. Then there exists a body-minimal pure Horn representation Ψ such that $\mathcal{B}(\Psi) \subseteq \mathcal{B}(\Phi)$.*

PROOF: Let X be a set determined in Step 4 of the algorithm and let $B \in \mathcal{B}(H_\Phi)$ be a body for which $X = F_H(B)$. Such a body exists according to the proof of Theorem 3. Then in Step 6 of the algorithm hyperedges $\{(B, v) : \ v \in Y - B\}$ could be added to H instead of $\{(X, v) : \ v \in Y - X\}$ where $Y = F_{H_\Phi}(X)$. Indeed, for every $v \in Y - X$, (B, v) is a valid hyperedge and covers every set that is covered by (X, v), proving the theorem. □

4 Guigues-Duquenne basis

In [9], a canonical body-minimal representation of pure Horn functions has been introduced called as **Guigues-Duquenne basis (GD basis)**. Algorithms for determining the GD basis of a pure Horn function h given by an arbitrary pure Horn CNF Φ were proposed in [3,7]. In what follows, we show that our algorithm also finds the GD basis.

The uniqueness of the GD basis lies in its saturation, a notion that has been introduced already in [2, 4]. A pure Horn CNF representation $\Phi = (V, \mathcal{C})$ is called **right-saturated** if for every clause $(B \to v) \in \mathcal{C}$ we have $(B \to v') \in \mathcal{C}$ for every $v' \in F_\Phi(B) \setminus B$, and is called **left-saturated** if $B_1 \subset B_2$ for $(B_1 \to v_1), (B_2 \to v_2) \in \mathcal{C}$ implies $v_1 \in B_2$. Finally, Φ is **saturated** if it is both left- and right-saturated.

These definitions can be naturally extended to directed hypergraphs: $H = (V, \mathcal{E})$ is **right-saturated** if $(B, v) \in \mathcal{E}$ implies $(B, v') \in \mathcal{E}$ for every $v' \in F_H(B) \setminus B$, and H is **left-saturated** if $(B_2 \subset B_1)$ for $(B_1, v_1), (B_2, v_2) \in \mathcal{E}$ implies $v_2 \in B_1$. Finally, H is **saturated** if it is both left- and right-saturated. It is easy to check that Φ is left- or right-saturated if and only if H_Φ is left- or right-saturated, respectively.

For sake of completeness, we prove that pure Horn functions have a unique saturated representation.

Theorem 5 *A pure Horn function has a unique saturated representation.*

PROOF: Assume indirectly that the pure Horn function h has two different saturated representations $H_1 = (V, \mathcal{E}_1)$ and $H_2 = (V, \mathcal{E}_2)$. Let (B, v) be a hyperdege in the symmetric difference of $\mathcal{E}_1$ and $\mathcal{E}_2$ with $|B|$ being minimal. Without loss of generality, assume that $(B, v) \in \mathcal{E}_1$. Then $B \notin \mathcal{B}(H_2)$ as otherwise H_2 is not right-saturated. As $B \in \mathcal{F}_h$, there exists a hyperedge $(B', w) \in \mathcal{E}_2$ covering B. By the choice of (B, v), we have $(B', u) \in \mathcal{E}_1$, thus H_1 is not left-saturated, a contradiction. □

Now we show that our algorithm determines the GD basis.

Theorem 6 *The output of Procedure* BodyMinimal *is the GD basis of h.*

Proof: By Theorem 5, it suffices to show that the output Ψ of the algorithm is both left- and right-saturated. Let $H = (V, \mathcal{E})$ denote the directed hypergraph constructed by the algorithm and let X be a body of Ψ. That is, $X = F_{H'}(B)$ for some $B \in \mathcal{B}(H_\Phi)$ where H' denotes the hypergraph constructed by the algorithm before considering X in Step 5. As H represents Φ, $F_{H_\Phi}(X) = F_H(X)$. Indeed, $F_{H_\Phi}(X)$ is the unique smallest true set in $\mathcal{T}_\Phi$ that contains X while $F_H(X)$ is the unique smallest true set in $\mathcal{T}_H$ containing X, hence they must coincide. Similarly, $F_{H_\Phi}(B) = F_H(B)$. But H' is a subhypergraph of H, hence $F_{H'}(B) = X$ implies $F_H(B) = F_H(X)$. Concluding these observations, we get $F_H(X) = F_{H_\Phi}(B)$. Step 7 of the algorithm ensures that $(X, v) \in \mathcal{E}$ for $v \in F_H(X) \setminus X$. Thus H, and in turn Ψ are indeed right-saturated.

Now consider two clauses $(X_1 \to v_1)$ and $(X_2 \to v_2)$ of Ψ such that $X_1 \subset X_2$. By Theorem 1, X_1 and X_2 must be independent false sets, hence there exists a true set $Y \in \mathcal{T}_{H_\Phi}$ separating X_1 and X_2, that is, $X_1 \subset Y \subset X_2$. As $\mathcal{T}_H$ and $\mathcal{T}_{H_\Phi}$ coincide, H may contain only hyperedges not covering any true set in $\mathcal{T}_{H_\Phi}$, hence $v_1 \in X_2$ as required. Thus H, and in turn Ψ are left-saturated. □

5 Edge-minimal representations

While trying to give a good approximation algorithm for finding an edge-minimal representation, we came to the following interesting result which may be useful in further examinations.

Theorem 7 *Assume that $H_1 = (V, \mathcal{E}_1)$ and $H_2 = (V, \mathcal{E}_2)$ are two hypergraph representations of a pure Horn function h. Then there exists a hypergraph representation $H = (V, \mathcal{E})$ of h such that $|\mathcal{E}| \leq |\mathcal{E}_1|$, $\mathcal{H}(H) \subseteq \mathcal{H}(H_1)$ and $\mathcal{B}(H) \subseteq \mathcal{B}(H_2)$.*

Proof: We may assume that for every hyperedge in $\mathcal{E}_1$ there exists a false set covered only by that hyperedge, as otherwise the hyperedge could be simply deleted.

Our proof is based on the following lemma.

Lemma 8 *For every hyperedge $(B, v) \in \mathcal{E}_1$ such that $B \notin \mathcal{B}(H_2)$, there exists a body $B' \in \mathcal{B}(H_2)$ such that $(V, \mathcal{E}_1 - (B, v) + (B', v))$ is also a representation of h.*

Proof: Let $(B, v) \in \mathcal{E}_1$ be a hyperedge of H_1 such that $B \notin \mathcal{B}(H_2)$ and let $M \subseteq V$ be an inclusion-wise maximal set such that $(V, \mathcal{E}_1 - (B, v) + (M, v))$ is also a representation of h. We distinguish two cases.

Case 1. $M \in \mathcal{B}(H_2)$
In this case $B' = M$ satisfies the requirements of the lemma.

Case 2. $M \notin \mathcal{B}(H_2)$
Note that M is a false set, hence there exists a hyperedge $h = (B', v') \in \mathcal{E}_2$ covering M. Let Y be the forward chaining closure of B', and let $B'' = M \cup Y$.

If $v \in Y$, hyperedge (B', v) is valid and covers all sets covered by (M, v). This means that $H' = (V, \mathcal{E}_1 - (B, v) + (B', v))$ is also a representation of h.

Assume now that $v \notin Y$. We claim that $H'' = (V, \mathcal{E}_1 - (B, v) + (B'', v))$ is a representation of h, contradicting the maximality of M. To prove this, it suffices to show that hyperedge (B'', v) covers all false sets that are covered only by (M, v) in $\mathcal{E}_1 - (B, v) + (M, v)$. Let F be such a false set and assume that it is not covered by (B'', v), that is, $Y \not\subseteq F$. Define $F' := F \cap Y$. By $B' \subseteq M \subseteq F$ and $B' \subseteq Y$, we have $B' \subseteq F'$. As Y is the forward chaining closure of B', F' is a false set. Hence H_1 has a hyperedge covering F'. As Y is a true set, this hyperedge has its body in F and head in $Y - F$, contradicting to our original assumption that F is covered only by (M, v), thus concluding the proof of the lemma. □

We can apply the lemma for each body in $\mathcal{B}(H_1) - \mathcal{B}(H_2)$, thus the theorem follows. □

A surprising corollary of the theorem, which also appeared in [1] in a completely different context, is as follows.

Corollary 9 *Every pure Horn function h has a representation which is both edge-minimal and body-minimal.*

Proof: Let H_1 and H_2 be edge minimal and body minimal representations of h, respectively. By applying Theorem 7 to H_1 and H_2, the resulting representation H is both edge and body minimal. □

The next corollary may serve as a starting point for approximating edge-minimal representations.

Corollary 10 *Every pure Horn function h ha an edge-minimal representation which is the subset of the GD-basis.*

Proof: Let H_1 be an edge-minimal representations of h let H_2 denote the GD basis. As H_2 is right-saturated, the hypergraph provided by Theorem 7 is a subhypergraph of H_2. □

Acknowledgement

The authors were supported by the MTA-ELTE Egerváry Research Group and by the Hungarian Scientific Research Fund - OTKA, No K109240.

References

[1] K. Adaricheva, J. B. Nation, On implicational bases of closure systems with unique critical sets, *Discrete Applied Mathematics*, 162:51–69, 2014.

[2] M. Arias, J. L. Balcázar, Query learning and certificates in lattices, In *Algorithmic Learning Theory*, pages 303–315. Springer, 2008.

[3] M. Arias, J. L. Balcázar, Canonical horn representations and query learning, In *Algorithmic Learning Theory*, pages 156–170. Springer, 2009.

[4] M. Arias, A. Feigelson, R. Khardon, R. A. Servedio, Polynomial certificates for propositional classes, *Information and Computation*, 204(5):816–834, 2006.

[5] G. Ausiello, A. DAtri, D. Sacca, Minimal representation of directed hypergraphs, *SIAM Journal on Computing*, 15(2):418–431, 1986.

[6] A. Bhattacharya, B. DasGupta, D. Mubayi, Gy. Turán, On approximate horn formula minimization, In: *ICALP, Lecture Notes in Computer Science*, (1), volume 6198: 438–450, 2010.

[7] E. Boros, O. Čepek, K. Makino, A combinatorial min-max theorem and minimization of pure-horn functions, In *International Symposium on Artificial Intelligence and Mathematics*, 2016.

[8] G. Gallo, G. Longo, S. Pallottino, S. Nguyen, Directed hypergraphs and applications, *Discrete Applied Mathematics*, 42(2):177–201, 1993.

[9] J.-L. Guigues, V. Duquenne, Familles minimales d'implications informatives résultant d'un tableau de données binaires, *Mathématiques et Sciences humaines*, 95:5–18, 1986.

[10] H. K. Büning, T. Lettmann, Propositional Logic: Deduction and Algorithms, Cambridge University Press, 1999.

[11] D. Maier, Minimum covers in relational database model, *Journal of ACM*, 27(4):664–674, 1980.

[12] M. Thakur, R. Tripathi, Linear connectivity problems in directed hypergraphs, *Theoretical Computer Science*, 410(27):2592–2618, 2009.

Blocking optimal structures

Kristóf Bérczi

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
berkri@cs.elte.hu

Attila Bernáth

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
athos@cs.elte.hu

Tamás Király

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
tkiraly@cs.elte.hu

Gyula Pap

MTA-ELTE Egerváry Research Group
Eötvös Loránd University
Budapest, Hungary
gyuszko@cs.elte.hu

Abstract: We consider weighted blocking problems (a.k.a. weighted transversal problems) of the following form. Given a finite set S, weights $w : S \to \mathbb{R}_+$, and a family $\mathcal{F} \subseteq 2^S$, find $\min\{w(H) : H \subseteq S,\ H \text{ intersects every member of } \mathcal{F}\}$.

In our problems S is the set of edges of a (directed or undirected) graph and $\mathcal{F}$ is the family of optimal solutions of a combinatorial optimization problem. In particular, we study the following four kinds of families:

- minimum cost k-spanning trees (unions of k edge-disjoint spanning trees)
- minimum cost s-rooted k-arborescences (unions of k arc-disjoint arborescences rooted at node s)
- minimum cost undirected k-braids between nodes s and t (unions of k edge-disjoint s-t paths)
- minimum cost directed k-braids between nodes s and t.

Note that the cost function $c : S \to \mathbb{R}_+$ that defines the family $\mathcal{F}$ and the weight function $w : S \to \mathbb{R}_+$ are not related. We consider the special cases where c or w is uniform. If $c \equiv 0$ (i.e. we want to block all combinatorial objects, not just the optimal ones), then most of the problems are NP-complete. However, if c is arbitrary but $w \equiv 1$ (a minimum cardinality transversal problem for $\mathcal{F}$), then our problems turn out to be polynomial-time solvable.

Keywords: minimum transversal, minimum weight transversal, k-spanning tree, k-arborescence, k-braid

1 Introduction

By **blocking problems** we mean the following type of problems. Given a finite set S and a family $\mathcal{F} \subseteq 2^S$, find $\min\{|H| : H \subseteq S,\ H \text{ intersects every member of } \mathcal{F}\}$. The family $\mathcal{F}$ encodes some optimal structures, for example minimum cost k-spanning trees of a graph (where S is the set of edges of a graph and a cost of each edge is given), or minimum cost k-arborescences of a digraph (where S is the set of arcs of a digraph and again we have a cost function on S). In the literature, these types of problems are also called **minimum transversal problems** for the family $\mathcal{F}$.

In a more general setting, we consider **weighted blocking problems** (or **minimum weight transversal problems**), that is, a weight function $w : S \to \mathbb{R}_+$ is also given and we want to find $\min\{w(H) : H$ intersects every member of $\mathcal{F}\}$. Note that this weight function is independent from the cost function that defines the family $\mathcal{F}$.

In particular, we will investigate the weighted blocking problem for four types of combinatorial structures: optimal k-spanning trees, optimal k-arborescences, and optimal undirected and directed k-braids. Let us define these objects.

Given an undirected graph $G = (V, E)$, a **k-spanning tree** is a subset B of edges that can be written as the union of k pairwise edge-disjoint spanning trees. It is known that k-spanning trees form the family of bases of a matroid, and that a minimum cost k-spanning tree can be found in polynomial time, if the cost $c(e)$ of each edge is given.

A **spanning arborescence** in a digraph $D = (V, A)$ is an arc set $F \subseteq A$ that is a spanning tree in the undirected sense and every node has in-degree at most one. Thus there is exactly one node, the **root node**, with in-degree zero. If the node set is clear from the context, spanning arborescences will be called **arborescences** for brevity. The arc-disjoint union of k spanning arborescences is called a **k-arborescence**. If every arborescence in the decomposition has the same root node s, then F is called an **s-rooted k-arborescence**. Given $D = (V, A)$, a positive integer k and a cost function $c : A \to \mathbb{R}_+$, a minimum cost k-arborescence or a minimum cost s-rooted k-arborescence can be found efficiently using the matroid intersection algorithm; see [14, Chapter 53.8] for a reference, where several related problems are considered. The existence of an s-rooted k-arborescence is characterized by Edmonds' disjoint arborescence theorem, while the existence of a k-arborescence is characterized by a theorem of Frank [6]. Frank also gave a linear programming description of the convex hull of k-arborescences, generalizing Edmonds' linear programming description of the convex hull of s-rooted k-arborescences.

Given an undirected graph $G = (V, E)$ and nodes s, t, an **undirected k-braid between s and t** is a subset of edges of G that can be decomposed into k pairwise edge-disjoint $s - t$ paths. Directed k-braids are defined analogously: in a digraph $D = (V, A)$, a **directed k-braid between nodes s and t** is a subset of arcs that can be decomposed into k pairwise arc-disjoint directed $s - t$ paths. We will use the term k-braid if we mean both the directed and undirected cases, or if the type of the graph is clear from context. Also, nodes s and t are omitted if they are clear from the context. It is known from network flow theory that we can find minimum cost (directed or undirected) k-braids if the non-negative cost $c(e)$ of every edge/arc is given.

Let us mention some known special cases of blocking problems. The **cuts** (or co-cycles) of a matroid are the minimal transversals of the family of bases; in other words, a subset of the elements is a cut if it is an inclusionwise minimal subset that contains at least one element from each basis. The problem of finding minimum cuts in matroids has been studied in several different contexts (note the distinction between *minimal* and *minimum*: minimal is shorthand for *inclusionwise minimal*, while minimum means *minimum size*). Perhaps the best known special case is the minimum cut problem in graphs, which can be solved using network flows, and faster algorithms have also been developed (e.g. the Nagamochi-Ibaraki algorithm [13]). This corresponds to the minimum cardinality blocking problem for spanning trees; moreover, these methods also find the minimum weight cut, so they solve the minimum weight blocking problem for spanning trees, too.

The minimum cut of kM, where M is a graphic matroid (or even a hypergraphic matroid, see [11]), can also be found in polynomial time. However, these methods do not extend to the minimum weight cut problem. Another notable open question is the complexity of finding a minimum cut in a rigidity matroid.

The minimum cut of a transversal matroid can also be found in polynomial time; however, the problem of finding a minimum *circuit* of a transversal matroid is NP-complete [12], which implies that the minimum cut problem is NP-complete for gammoids. Another line of research considers the problem for binary matroids. NP-completeness was proved by Vardy [16]; Geelen, Gerards, and Whittle [8] conjecture that the problem is in P for any minor-closed proper subclass of binary matroids. Partial results in this direction have been achieved by Geelen and Kapadia [9].

If we consider *minimum cost bases* (or *optimal bases* for brevity) of a matroid M, then these form

the bases of another matroid which can be obtained by taking the direct sum of certain minors of M. Thus we can find a minimum transversal of the family of optimal bases of M by solving minimum cut problems in some minors of M. In particular, if the minimum cut problem is solvable in polynomial time in a minor-closed class of matroids, then a minimum transversal of optimal bases can also be found in polynomial time in this class. For example, since the class of graphic matroids is minor-closed and the minimum cut problem can be solved efficiently, we can also efficiently find a minimum transversal of optimal spanning trees in a graph with edge costs.

Arborescences can be considered as common bases of two matroids, so the problem of finding a minimum transversal of the family of arborescences is a special case of the minimum transversal problem for common bases of two matroids. This problem is NP-hard in general (as mentioned above, it is NP-hard even when the two matroids coincide). However, the special case for arborescences can be formulated as the minimization of the sum of the in-degrees of two disjoint node sets of the digraph, which can be solved efficiently using network flows. The problem of finding a minimum transversal of the family of *minimum cost arborescences* is considerably more difficult. It can still be solved in polynomial time as shown in [4], but the solution requires more sophisticated tools than network flows.

In this paper we consider the following problems.

Problem 1 *Given a graph $G = (V, E)$, cost function $c : E \to \mathbb{R}_+$, weight function $w : E \to \mathbb{R}_+$, and a positive integer k, find* $\min\{w(H) : H$ *intersects every c-optimal k-spanning tree in $G\}$.*

Problem 2 *Given a digraph $D = (V, A)$, cost function $c : A \to \mathbb{R}_+$, weight function $w : A \to \mathbb{R}_+$, node $s \in V$ and a positive integer k, find* $\min\{w(H) : H$ *intersects every c-optimal s-rooted k-arborescence$\}$.*

Problem 3 *Given a graph $G = (V, E)$, cost function $c : E \to \mathbb{R}_+$, weight function $w : E \to \mathbb{R}_+$, nodes $s, t \in V$ and a positive integer k, find* $\min\{w(H) : H$ *intersects every c-optimal k-braid from s to $t\}$.*

Problem 4 *Given a digraph $D = (V, A)$, cost function $c : A \to \mathbb{R}_+$, weight function $w : A \to \mathbb{R}_+$, nodes $s, t \in V$ and a positive integer k, find* $\min\{w(H) : H$ *intersects every c-optimal k-braid from s to $t\}$.*

We consider two types of restrictions on w and c. When $w \equiv 1$, i.e. w is uniform, our problems are **minimum cardinality transversal problems**, and they turn out to be polynomial-time solvable. The second type of restriction is $c \equiv 0$, that is, we want to **block all combinatorial objects**, not just the optimal ones. Note that $c \equiv 1$ could also be chosen for k-spanning trees or k-arborescences, but it is not suitable to describe all k-braids. With this restriction, most of our problems are NP-complete. We leave two questions open: we do not know the status of Problem 1 even for $c \equiv 0$ and $k = 2$, and we do not know the status of Problem 2 if $w \equiv 1$. Our results are summarized in Table 1 below.

	Uniform weight ($w \equiv 1$)	**Uniform cost** ($c \equiv 0$)
Blocking optimal k-spanning trees (Problem 1)	polynomial (Theorem 8)	Open (open even for $k = 2$)
Blocking optimal k-arborescences (Problem 2)	Open polynomial for fixed k [2] polynomial if $c \equiv 0, w \equiv 1$ [3]	NP-complete (Theorem 12) (polynomial for fixed k [3])
Blocking optimal undirected k-braids (Problem 3)	polynomial (Theorem 16)	NP-complete (Theorem 18) (polynomial for fixed k, see Section 4)
Blocking optimal directed k-braids (Problem 4)	polynomial (Theorem 14)	NP-complete (Theorem 18) (polynomial for fixed k, see Section 4)

Table 1: Summary of results

1.1 Notation

Let us overview some of the notation and definitions used in the paper. A **partition** $\mathcal{P}$ of a set V is a collection of pairwise disjoint non-empty subsets of V that together cover V. The partition is **trivial** if it consists of the single set V. We will use the notation $|\mathcal{P}|$ to mean the number of sets in the partition $\mathcal{P}$. A set family $\mathcal{L} \subseteq 2^V$ is said to be **laminar** if any two members of $\mathcal{L}$ are either disjoint, or one contains the other. For a function $x : A \to \mathbb{R}$ and subset $Z \subseteq A$, we use the notation $x(Z) = \sum_{a \in Z} x_a$.

Given a (directed or undirected) graph $G = (V, E)$ and some $W \subseteq V$, let $G[W] = (W, \{uv \in E : u, v \in W\})$ be the restriction of G to W, and G/W be the graph that we obtain from G by contracting W into a single node (and deleting the loops that arise). If $B \subseteq E$ then we will also use $B[W]$ to mean the restriction of (V, B) to W and B/W to mean the contraction of W in (V, B). If $H \subseteq E$ then $G - H = (V, E - H)$ is the graph obtained from G by deleting the edges in H. Furthermore, if $\mathcal{L} \subseteq 2^V$ is a laminar family and $W \in \mathcal{L}$, then we denote by $\mathcal{L}/W$ the laminar family that is obtained from $\mathcal{L}$ by contracting W into a single node.

For a graph $G = (V, E)$ and some $Z \subseteq V$, $\delta_G(Z)$ denotes the set of edges in E with exactly one end-node in Z, and $d_G(Z) = |\delta_G(Z)|$ is the number of these edges. A graph G is said to be **k-edge-connected** if $d_G(Z) \geq k$ for every $\emptyset \neq Z \subsetneq V$.

2 Blocking optimal k-spanning trees

For a graph $G = (V, E)$ and a partition $\mathcal{P}$ of the nodes of G, we denote by $e_G(\mathcal{P})$ the number of edges of G that go between two different classes of $\mathcal{P}$ (**cross-edges** in the partition $\mathcal{P}$).

Definition 5 *An undirected graph G is said to be (k, l)-**partition-connected** if $e_G(\mathcal{P}) \geq k(|\mathcal{P}| - 1) + l$ holds for any non-trivial partition $\mathcal{P}$ of the nodes of G.*

Theorem 6 (Tutte, [15]) *A graph contains k edge-disjoint spanning trees if and only if it is $(k, 0)$-partition-connected.*

Theorem 7 *Given a graph $G = (V, E)$ and two positive integers k, l, we can decide in polynomial time if G is (k, l)-partition-connected or not. If it is not, then one can also find a partition $\mathcal{P}$ satisfying $e_G(\mathcal{P}) < k(|\mathcal{P}| - 1) + l$.*

Proof: If $l \geq k$ then G is (k, l)-partition-connected if and only if it is $k + l$-edge-connected. This can be checked in polynomial time with network flows, or the Nagamochi-Ibaraki minimum cut algorithm [13]. If the graph is not (k, l)-partition-connected then a minimum cut can serve as a partition with 2 classes as a witness.

On the other hand, if $l \leq k$ then the solution is described in [7], page 305. □

Using these results, we can solve Problem 1 in polynomial time in the special case when both c and w are uniform ($w \equiv 1$ and $c \equiv 0$): simply find (by logarithmic search) the smallest positive integer l such that G is not (k, l)-partition-connected, along with a partition $\mathcal{P}$ satisfying $e_G(\mathcal{P}) < k(|\mathcal{P}|-1)+l$. The optimal solution will be an arbitrary subset of cross-edges of $\mathcal{P}$ of size l (note that $e_G(\mathcal{P}) = k(|\mathcal{P}| - 1) + l - 1 \geq l$, as G is $(k, l - 1)$-partition-connected). This approach can be extended to deal with the case where c is not uniform.

Theorem 8 *Problem 1 is solvable in polynomial time if $w \equiv 1$.*

Proof: From the dual characterization of optimal k-spanning trees we get the following lemma.

Lemma 9 *Given a graph $G = (V, E)$, positive integer k and a cost function $c : E \to \mathbb{R}_+$, we can find in polynomial time disjoint subsets $E_0, E_1 \subseteq E$ and a laminar family $\mathcal{L} \subseteq 2^V$ so that for any k-spanning tree $B \subseteq E$ the following statements are equivalent:*

1. *B is a c-optimal k-spanning tree,*

2. *$E_1 \subseteq B \subseteq E - E_0$ and $B[W]$ is a k-spanning tree of $G[W]$ for every $W \in \mathcal{L}$.* □

We say that E_0 is the set of **forbidden edges**, while E_1 is the set of **mandatory edges**. Moreover, given a graph $G = (V, E)$ and a laminar family $\mathcal{L} \subseteq 2^V$, we say that a k-spanning tree $B \subseteq E$ is **$\mathcal{L}$-tight** if $B[W]$ is a k-spanning tree of $G[W]$ for every $W \in \mathcal{L}$. Note that $B \subseteq E$ is an $\mathcal{L}$-tight k-spanning tree if and only if it can be decomposed into k edge-disjoint $\mathcal{L}$-tight spanning trees. For later reference we state the following problem.

Problem 10 (Blocking $\mathcal{L}$-tight k-spanning trees) *Given a graph $G = (V, E)$ and a laminar family $\mathcal{L} \subseteq 2^V$, find $\min\{|H| : H$ intersects every $\mathcal{L}$-tight k-spanning tree$\}$.*

Lemma 9 implies that the problem of blocking optimal k-spanning trees (Problem 1 for $w \equiv 1$) can be reduced to the problem of blocking $\mathcal{L}$-tight k-spanning trees. Indeed, if there are mandatory edges then we can block all optimal k-spanning trees by a single (mandatory) edge. Otherwise, we can just remove the forbidden edges, and the problem is to block $\mathcal{L}$-tight k-spanning trees in $G - E_0$. The rest of the proof is about the solution of Problem 10. We note that we can decide in polynomial time if an $\mathcal{L}$-tight k-spanning tree exists at all: this is a maximum cost k-spanning tree problem by setting the cost of an edge $uv \in E$ to be the number of sets in $\mathcal{L}$ that contain both endpoints of the edge, that is $cost(uv) = |\{W \in \mathcal{L} : u, v \in W\}|$.

The following observation leads us to the solution of Problem 10.

Claim 11 *Given a graph $G = (V, E)$ and a laminar family $\mathcal{L} \subseteq 2^V$, let $W \in \mathcal{L}$ be an inclusionwise minimal member of $\mathcal{L}$. A subset $B \subseteq E$ is an $\mathcal{L}$-tight k-spanning tree if and only if $B[W]$ is a k-spanning tree in $G[W]$, and B/W is an $\mathcal{L}/W$-tight k-spanning tree in G/W.*

Proof: Clearly, if $B \subseteq E$ is an $\mathcal{L}$-tight k-spanning tree then $B[W]$ is a k-spanning tree in $G[W]$, and B/W is an $\mathcal{L}/W$-tight k-spanning tree in G/W. On the other hand, if $B \subseteq E$ satisfies that $B[W] = \dot{\bigcup}_{i=1}^{k} F_i^1$ and $B/W = \dot{\bigcup}_{i=1}^{k} F_i^2$ where each F_i^1 is a spanning tree of $G[W]$ and each F_i^2 is an $\mathcal{L}/W$-tight spanning tree in G/W then we can simply set $F_i = F_i^1 \cup F_i^2$ for $i = 1, \ldots, k$ and obtain that $B = \dot{\bigcup}_{i=1}^{k} F_i$ is an $\mathcal{L}$-tight k-spanning tree in G, as F_i is an $\mathcal{L}$-tight spanning tree in G. •

Using this claim, the solution of Problem 10 is the following. Pick an inclusionwise minimal member W of $\mathcal{L}$ and solve the problem of blocking all k-spanning trees in $G[W]$ (as described after Theorem 7) to get a candidate. Then recursively solve the problem of blocking $\mathcal{L}/W$-tight k-spanning trees in G/W. Finally, output the best of the candidates found during the algorithm. □

As an open problem we pose the following question: can we solve Problem 1 in polynomial time if c is uniform but w is not, that is, given a graph $G = (V, E)$, a positive integer k and $w : E \to \mathbb{R}_+$, can we determine $\min\{w(H) : H \subseteq E, G - H$ does not admit a k-spanning tree $\}$? We do not know how to solve this problem even for fixed k, e.g. $k = 2$.

Remark. For $k = 1$, the above problem is equivalent to the minimum weight cut problem: given a graph $G = (V, E)$ and $w : E \to \mathbb{R}_+$, find $\min\{w(H) : H \subseteq E,\ G - H$ is not connected$\}$. This problem has another extension for larger k, namely the following **k-edge-connectivity blocking problem**: given a graph $G = (V, E)$ and $w : E \to \mathbb{R}_+$, find $\min\{w(H) : H \subseteq E,\ G - H$ is not k-edge-connected$\}$. Note that this problem cannot be formulated as the weighted blocking of some optimal structures, however it is related to (the uniform cost version of) both Problems 1 and 3. We claim that this problem is solvable in polynomial time, even if k is part of the input. The algorithm is analogous to the algorithm for connectivity interdiction developed by Zenklusen [17]; here we only sketch the proof. (In contrast, Problem 3 with $c \equiv 0$ is NP-complete, see Theorem 18.)

Let $e_1, e_2, \ldots, e_m$ be the enumeration of the edges ordered by increasing weight, and let $E_i = \{e_1, \ldots, e_i\}$. For every $i \in [m]$, we solve the following problem: find $\min\{w(\delta_G(Z) \cap E_i) : \emptyset \neq Z \subsetneq V, |\delta_G(Z) \setminus E_i| \leq k-1\}$. This is a bicriteria minimum cut problem, that can be solved in polynomial time using the method of Armon and Zwick [1]. Let $\ell \in [m]$ be the index for which the minimum is the smallest, and let Z be the core of the corresponding cut. We claim that $H = \delta_G(Z) \cap E_\ell$ is the optimal solution of the blocking problem. On one hand, removing H results in a graph that is not k-edge-connected because $|\delta_G(Z) \setminus H| \leq k-1$. On the other hand, if H' is an optimal solution of the blocking problem, then there is a subset Z' such that H' contains the $d_G(Z') - k + 1$ edges with the smallest weight from $\delta_G(Z')$. Thus $H' = \delta_G(Z') \cap E_i$ for some i, and $|\delta_G(Z') \setminus E_i| \leq k-1\}$. It follows that $w(H') = w(\delta_G(Z') \cap E_i) \geq w(\delta_G(Z) \cap E_\ell) = w(H)$, so H is also optimal.

3 Blocking optimal k-arborescences

Problem 2 for $k = 1$ was solved in [4]. For $w \equiv 1$, an algorithm solving Problem 2 was given in [2] that has polynomial running time if k is fixed. If both w and c are uniform, then the problem is polynomially solvable, as was shown in [3]. Furthermore, it was observed in [3] that for uniform c and fixed k the problem is solvable in polynomial time (with a simple brute force technique). In this light it is perhaps surprising that Problem 2 is NP-complete for $c \equiv 0$, if k is part of the input.

Theorem 12 *Problem 2 is NP-complete in the special case $c \equiv 0$.*

The proof of this theorem will be given later, together with the proof of Theorem 18.

4 Blocking optimal k-braids

Our solution for the minimum cardinality blocking of optimal directed and undirected k-braids (Problems 3 and 4 for $w \equiv 1$) is based on the following result.

Theorem 13 (Ford and Fulkerson [5]) *Given a digraph $D = (V, A)$, $s, t \in V$, $k \in \mathbb{Z}_+$ and a cost function $c : A \to \mathbb{R}_+$, the minimum cost of a directed k-braid from s to t is equal to*

$$\max\left\{k\pi(t) + \sum[c_\pi(uv) : uv \in A,\ c_\pi(uv) < 0] : \pi \in \mathbb{R}_+^V,\ \pi(s) = 0\right\}, \tag{1}$$

where $c_\pi(uv) = c(uv) - \pi(v) + \pi(u)$ for an arc $uv \in A$.

Based on Theorem 13, first we show how the minimum cardinality blocking of c-optimal directed k-braids can be solved in polynomial time. The undirected case is then reduced to the directed one.

Theorem 14 *Problem 4 is solvable in polynomial time in the special case $w \equiv 1$.*

Proof: Choose an optimal solution π^* of (1) and let $A_- = \{uv \in A : c_{\pi^*}(uv) < 0\}$, $A_0 = \{uv \in A : c_{\pi^*}(uv) = 0\}$, and $A_+ = \{uv \in A : c_{\pi^*}(uv) > 0\}$. Note that π^* and thus A, A_0, A_+ can be found in polynomial time (see eg. [7, Theorem 3.6.1]). We refer to members of A_0 as **tight arcs**.

The complementary slackness conditions imply that a k-braid $F \subseteq A$ is optimal if and only if $A_- \subseteq F \subseteq A - A_+$. Hence, the problem of blocking optimal directed k-braids can be solved as follows.

Case 1: $A_- \neq \emptyset$. In this case the optimal k-braids can be blocked by a single arc from A_-.

Case 2: $A_- = \emptyset$**.** In this case an optimal solution consists of all-but-$(k-1)$ arcs from a minimum $s-t$ cut in $D_0 = (V, A_0)$. That is, the minimum number of arcs blocking all c-optimal directed k-braids is

$$\min\{\varrho_{A_0}(Z) - (k-1) : t \in Z \subseteq V - s\}.$$

This concludes the proof of the theorem. □

In order to deal with the undirected case, we need a lemma on inclusionwise minimal transversals of optimal directed k-braids.

Lemma 15 *Given a digraph $D = (V, A)$, $s, t \in V$, $k \in \mathbb{Z}_+$ and a cost function $c : A \to \mathbb{R}_+$, if $H \subseteq A$ is an inclusionwise minimal arc set that intersects every optimal k-braid then H does not contain a directed cycle.*

PROOF: Suppose, for contradiction, that there exists a directed cycle $C = \{f_1, \ldots, f_\ell\}$ in H. By the minimality of H, there exists a c-optimal k-braid B_i with $B_i \cap H = \{f_i\}$ for $i = 1, \ldots, \ell$. Let $B = \cup_{i=1}^{\ell} B_i - C$ and define a capacity function $g : B \to \mathbb{Z}_+$ by setting $g(f) = |\{i : \ f \in B_i\}|$. Note that $g(f) \leq \ell$ for every arc $f \in B$.

We claim that we can pack ℓ k-braids $B'_1, \ldots, B'_\ell$ in B under the capacities. Indeed, it is known (see e.g. [14, (13.12)]) that the convex hull P of incidence vectors of those subsets of B that contain k arc-disjoint $s-t$ paths is determined by

$$\begin{aligned} 0 \leq x(a) &\leq 1 \qquad \text{for each } a \in B, \\ x(C) &\geq k \qquad \text{for each } s-t \text{ cut } C. \end{aligned}$$

By a result of L.E. Trotter [14, Theorem 13.8], this polytope has the so-called integer decomposition property, meaning that for each $\ell \in \mathbb{Z}_+$, any integer vector $x \in \ell \cdot P$ is the sum of ℓ integer vectors in P. Clearly, $g \in \ell \cdot P$, hence the existence of $B'_1, \ldots, B'_\ell$ follows.

By the optimality of the B_is, $\sum_{i=1}^{\ell} c(B_i) = c(C) + \sum_{i=1}^{\ell} c(B'_i) \geq \sum_{i=1}^{\ell} c(B'_i) \geq \sum_{i=1}^{\ell} c(B_i)$. Thus equality must hold throughout and so B'_i is c-optimal for $i = 1, \ldots, \ell$, contradicting the assumption that H is a blocking arc-set. □

Now we turn to the problem of blocking undirected optimal k-braids.

Theorem 16 *Problem 3 is solvable in polynomial time in the special case $w \equiv 1$.*

PROOF: We will reduce Problem 3 to Problem 4. Consider an instance $G = (V, E)$ of Problem 3. We define a digraph $G^\circ = (V, E^\circ)$ and a cost function $c^\circ : E^\circ \to \mathbb{R}_+$ as follows: for each edge $e = uv$ of G, add a pair of symmetric arcs $e' = uv$ and $e'' = vu$ to E° with cost $c^\circ(e') = c^\circ(e'') = c(e)$. Denote the minimum size of a blocking set in G and G° by τ and τ°, respectively.

Lemma 17 $\tau = \tau^\circ$

PROOF: Let $H^\circ \subseteq E^\circ$ be an optimal solution in G°, that is, $|H^\circ| = \tau^\circ$ and H° covers every c°-optimal k-braid in G°. Let $H = \{uv \in E : uv \text{ or } vu \in H^\circ\}$. Clearly, H covers every c-optimal k-braid in G and $|H| \leq |H^\circ|$, hence $\tau \leq \tau^\circ$.

To see the other direction, take an optimal blocking set $H \subseteq E$ in G, that is, $|H| = \tau$ and H covers every c-optimal k-braid in G. Let $H^\circ = \{uv, vu \in E^\circ : \ uv \in H\}$. Now H° covers every c°-optimal k-braid in G°. Note that $|H^\circ| = 2|H|$ as H° contains both e' and e'' for each $e \in H$. However, by Lemma 15, H° contains a minimal blocking set that contains at most one of e' and e'' for each $e \in H$. This shows $\tau \geq \tau^\circ$, thus concluding the proof of the lemma. •

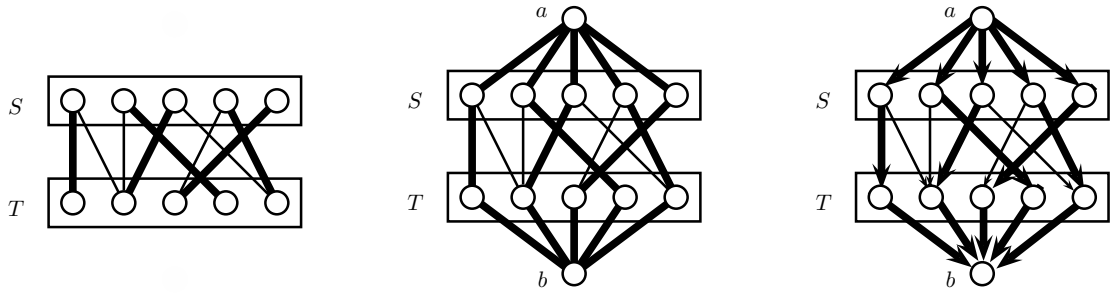
The theorem follows from Theorem 14 and Lemma 17. □

In contrast to the polynomial-time solvability of the minimum cardinality blocking problem of minimum cost k-braids, the weighted blocking problems for k-braids are NP-complete, even if $c \equiv 0$.

Theorem 18 *Problems 3 and 4 are both NP-complete in the special case $c \equiv 0$.*

PROOF OF THEOREMS 12 AND 18: Clearly, the decision versions of the mentioned problems are in NP, therefore we will only concentrate on proving their completeness.

Given a bipartite graph $G_0 = (S, T, E_0)$ with $|S| = |T| = n$, consider the following constructions (see Figure 1).



(a) Original graph (b) Graph G (c) Digraph D_1

Figure 1: Constructions for G and D_1

1. Let $a, b \notin S \cup T$ be new nodes and let $V = S \cup T \cup \{a, b\}$. Let $G = (V, E)$ where $E = E_0 \cup \{as : s \in S\} \cup \{tb : t \in T\}$.
2. Let $D_1 = (V, A_1)$ be the digraph obtained from G (defined in the previous paragraph) by orienting each edge "from a to b", that is $A_1 = \{as : s \in S\} \cup \{st : st \in E_0\} \cup \{tb : t \in T\}$.
3. Let $D_2 = (V, A_2)$ be obtained from D_1 (defined in the previous paragraph) by adding n parallel arcs from b to every $v \in S \cup T$.

Claim 19 *The following statements are equivalent.*

(i) *G_0 admits a perfect matching.*

(ii) *There is an undirected n-braid from a to b in G.*

(iii) *There is a directed n-braid from a to b in D_1.*

(iv) *There is an a-rooted n-arborescence in D_2.*

PROOF: It is quite straightforward how (i) implies all of the other items in the list above (see Figure 1 for an illustration). On the other hand, if G_0 does not have a perfect matching, then by Hall's theorem there is a subset $X \subseteq S$ with $|\Gamma_{G_0}(X)| < |X|$, and then the set $a + X + \Gamma_{G_0}(X)$ defines a cut that shows that neither of (ii)-(iv) can hold. •

The proof of the theorem can be finished as follows. We will reduce the following problem.

Problem 20 (Blocking Bipartite Matchings) *Given a bipartite graph $G_0 = (S, T, E_0)$, find* $\min\{|H| : H \subseteq E_0,\ G - H$ *does not have a perfect matching*$\}$.

It is known (see e.g. [10]) that Problem 20 is NP-complete. Given an instance $G_0 = (S, T, E_0)$ of this problem, we construct the graph G and digraphs D_1 and D_2 as above, and set the weights as follows. The (directed or undirected) edges $st \in E_0$ have weight 1, while any other edge has a large weight M (for example $M = |E_0| + 1$ suffices). Thus we have defined an instance of Problem 3 with input G, an

instance of Problem 4 with input D_1 and an instance of Problem 2 with input D_2: in all three cases we have also defined weights for edges/arcs, and the cost function is defined to be zero in all three cases. Then the problem of Blocking Bipartite Matchings in G_0 has a solution of size m if and only if either of the defined weighted blocking problems has a solution of total weight at most m. □

Note that both Problems 3 and 4 can be solved in polynomial time if $c \equiv 0$ and k is fixed, using a brute-force search technique (similar to the one used in [3] for solving Problem 2 for fixed k and $c \equiv 0$).

Acknowledgement

The research was supported by the Hungarian National Research, Development and Innovation Office – NKFIH, grants K109240 and K120254.

References

[1] A. Armon and U. Zwick. Multicriteria Global Minimum Cuts *Algorithmica* **46: 15** (2006)

[2] A. Bernáth and T. Király. Blocking optimal k-arborescences. *Technical Report, Egerváry Research Group, Budapest*, **TR-2015-09,** (2015) `www.cs.elte.hu/egres`.

[3] A. Bernáth and G. Pap. Blocking unions of arborescences. *Discrete Optimization*, **22, Part B:277 – 290**, (2016)

[4] A. Bernáth and G. Pap. Blocking optimal arborescences. *Mathematical Programming*, **161(1):583–601**, (2017)

[5] L. R. Ford and D. R. Fulkerson. *Flows in networks.* (1962)

[6] A. Frank. On disjoint trees and arborescences. In *Algebraic Methods in Graph Theory, Colloquia Mathematica Soc. J. Bolyai*, **volume 25, pages 159–169**, (1978)

[7] A. Frank. *Connections in Combinatorial Optimization. Oxford Lecture Series in Mathematics and Its Applications. OUP Oxford,* (2011)

[8] J. Geelen, B. Gerards, and G. Whittle. The highly connected matroids in minor-closed classes. *Annals of Combinatorics*, pages 1–17, (2013)

[9] J. Geelen and R. Kapadia. Computing girth and cogirth in perturbed graphic matroids. *arXiv preprint* **arXiv:1504.07647**, (2015)

[10] G. Joret and A. Vetta. Reducing the rank of a matroid. *Discrete Mathematics & Theoretical Computer Science*, **Vol. 17 no.2**, Jan. (2015)

[11] T. Király. Computing the minimum cut in hypergraphic matroids. *Technical Report, Egerváry Research Group, Budapest,* **QP-2009-05,** (2009) `www.cs.elte.hu/egres`.

[12] S. T. McCormick. A combinatorial approach to some sparse matrix problems. *Technical report, DTIC Document,* (1983)

[13] H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM Journal on Discrete Mathematics*, **5(1):54–66,** (1992)

[14] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, **Volume 24.** Springer Verlag, (2003)

[15] W. T. Tutte. On the problem of decomposing a graph into n connected factors. *Journal of the London Mathematical Society*, **1(1):221–230**, (1961)

[16] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, **43(6):1757–1766,** (1997)

[17] R. Zenklusen. Connectivity interdiction. *Operations Research Letters*, **42(67):450 – 454**, (2014)

Designing chess pairing mechanisms

Péter Biró[1]

Institute of Economics, Research Centre for Economic and Regional Studies, Hungarian Academy of Sciences, H-1112, Budaörsi út 45, Budapest, Hungary, and Department of Operations Research and Actuarial Sciences, Corvinus University of Budapest
peter.biro@krtk.mta.hu

Tamás Fleiner[2]

Department of Computer Science and Information Theory
Budapest University of Technology and Econimics
1117 Budapest, Magyar tudósok körútja 2., Hungary
fleiner@cs.bme.hu

Richárd Palincza

Department of Computer Science and Information Theory
Budapest University of Technology and Econimics
1117 Budapest, Magyar tudósok körútja 2., Hungary
richard.palincza@gmail.com

Abstract: The Swiss system is the most popular chess tournament system that is recognised and regulated by the World Chess Federation (FIDE). Chess pairings in each round of a Swiss tournament are conducted by sophisticated matching algorithms. The matching mechanisms are precisely defined in the FIDE guidebook [3], currently four different variants are allowed. The descriptions of the matching procedures are such that every arbiter should be able to conduct the pairings, even without computer assistance. However, many parts of these procedures are very inefficient, as they may terminate in highly exponential time in the number of players due to their exhaustive search nature. We demonstrate how the main priority rules of the Dutch variant can be replaced by efficient matching algorithms. These efficient algorithms can serve as the base of software tools used for pairings.

Keywords: maximum weight matching, mechanism design, roommates problem, tournaments

1 Introduction

The Swiss system is a pairing system invented by Dr. Julius Müller of Brugg, Switzerland. It was first used in a chess tournament at Zurich in 1895. It has been used in the Unites States since 1942 and also the team chess world championship, the so-called Olympiad was first organised with the Swiss system in Buenos Aires in 1978. The World Chess Federation (FIDE) allow currently four variants of the Swiss system to be used in individual tournaments, namely the Dutch system, Lim, Dubov and Burnstein systems. In this paper we focus on the Dutch system, which is the most classical one among the four.

[1]Research is supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2016-3/2016), and by OTKA grant no. K108673.
[2]Research is supported by OTKA grant no. K108383.

The corresponding matching procedures are precisely described in the FIDE guidebook [3], with a new guidance published very recently on the Dutch system to be implied from 1 July 2017. We note that the latest version of the Dutch system is much clearer than the previous one, the requirements and the priorities on the aimed pairing are described in a mathematically more structured way. We include the most important part of the pairing rule in the Appendix.

Yet, these matching procedures are still rather complicated and their descriptions contain some exhaustive search routines which can make the pairing very slow, i.e. highly exponential in the number of players. In this research we investigate whether some of these routines can be replaced by efficient matching algorithms, e.g. by the maximum size and maximum/minimum weight matching algorithms of Edmonds. A similar investigation has been done by Ólafsson [5], but his study was concerned with an older and simpler version of the Swiss system. In particular, the transposition and exchanges rules were not considered in his work, which are two highly exponential routines in the current versions (both in the currently used rule and also in the new one valid from 1 July 2017).

Finally, let us note that even if the translation of the current rules to efficient algorithms is possible for some routines, it can be still reasonable to keep the exhaustive search descriptions in the official descriptions as some arbiters might still do the pairings by hand (and it would be too demanding for the arbiters if FIDE would request them to conduct the Edmonds' algorithm by hand). However, in most tournaments the pairing are conducted by some software, and in their pairing algorithm it would be indeed useful to replace the inefficient routines with efficient algorithms. By our mathematical investigation we also aim to understand the priorities used in the matching process which implies the properties of the matchings obtained, as these are not obvious from the current descriptions of the matching rules.

First we describe the basic notions, rules and goals of the pairings, and then we investigate the particular routines used in the Dutch variant.

2 Basic description of the Swiss pairings

In this section we summarise the basic features of the Swiss pairings, the concepts, definition, rules and common priorities used in all of the four variants.

2.1 The basic rules and goals

The general goal of a chess tournament is to select the winner and rank the others. The most important requirement of the pairing is that everyone should play in each round of the tournament (except one player if the number of players is odd). Thus, in mathematical terms, the matching obtained should be (almost) complete. In the case of odd number of players one player will remain unmatched in every round, and gets a bye (i.e. 1 point) without colour. This cannot happen twice with any player during a tournament.

The second common criterion in the Swiss tournament is that no pair of players can play twice. Thus when considering a pairing problem these pairs are not eligible.

Finally, since playing a game with white or black can have significant effect on the result, a tournament is considered fair if every player has played approximately the same number of times with white and black. These colour rules are a bit softer and used differently in the four variants, but a common strict rule is not to let any player to play with the same colour three times in a row, and also not to let any player to have a colour difference greater than two. There are however some exceptions with regard to these rules in the very last round of a tournament.

2.2 Implementation of the goals

The basic concept of the Swiss pairing is to rank the players after each tournament according to their scores, and to match them from the strongest ones to the weakest ones sequentially according to their scores. To understand the procedure, first we have to describe the scoring rules of an individual chess tournament for the general readership. In a chess tournament with individual players a player gets score

1 if she wins, score half is she draws and zero score if she looses. A typical Swiss tournament has nine rounds, and after that the players are primarily ranked according to their total scores. The pairing procedure considers the players according to their scores and tries to match everyone to someone with the same score, starting with the strongest players. Thus after, say, five rounds of a Swiss tournaments, the process is to consider those players first those who have 5 points, and then of those with 4.5 points, and so on. The pairing process is described for players within each score-group.

Sometimes it is not possible to match everyone within the score group, as not all the pairs are eligible and also because we may have an odd number of players. In this case the task is to match as many players as possible, and the rest will be moved to the subsequent score group. These players are called *downfloaters*. In order to avoid the further downgrade of downfloaters when considering the subsequent score-group, the downfloaters must be all matched, if possible. Their opponents will be called *upfloaters*. As a weak rule, we may also want to avoid to select the same players to become down- or upfloaters, so we shall try not to give an identical float to any player in two consecutive rounds or twice in three consecutive rounds.

2.3 Mathematical notions

We describe the pairing problem of a chess tournament as a matching problem in a nonbipartite graph $G(N, E)$ with node set $N = \{1, 2, \dots n\}$ and edges set $E = \{e_1, e_2, \dots, e_m\}$. The players correspond to nodes and we have an edge $ij \in E(G)$ if players i and j are eligible to play with each other. A *matching* is a set of independent edges in E, i.e. every node is incident with at most edge in a matching. An *(almost) complete* matching has size $\lfloor \frac{n}{2} \rfloor$ in G.

Finding a maximum size or maximum weight matching in a nonbipartite graph can be done efficiently by Edmonds' algorithm. The best implementation for the maximum size algorithm has running time $O(\sqrt{n} \cdot m)$ according to [6], and the best currently known running time for the maximum weight matching algorithm is $O(nm + n^2 \log n)$ due to [2].

The selection of the pairing is based on various priority rules. As we will demonstrate, finding the right pairings can be done by using exponentially decreasing weights, or equivalently to do the optimisation with *weight-vectors* on the edges. We choose the latter technique to make the description simpler. In particular, for each edge $e = ij$ we will introduce a weight vector w_e of $O(n)$ length. The implementation of the pairing rule will be equivalent to finding a matching on the graph with a lexicographically maximal weight. Note that using the weight-vectors will increase the running time of the classical Edmonds algorithm by a factor of n, but still remains strongly polynomial in the number of players (n).

3 The Dutch system

The official description of the currently used Dutch system describes the subroutine used in the most inner cycle of the pairing process, and the exhaustive search method are then extended for the case when no ideal pairing is possible. However, in the new description (to be applied from 1 July 2017) these subroutines are only suggested to use after satisfying the main criteria and goals. Thus we will also follow the new description (partly included in the Appendix) and first describe the main criteria and then the transposition and exchange rules. After providing the short description we show how the subroutines can be implemented with efficient matching algorithms.

3.1 Summary of the FIDE description

Suppose that we consider a score-group S during the pairing process. If this is not the highest score-group then there may be some downfloaters, denoted by F. First we divide S into two subgroups $S1$ and $S2$ of approximately the same size, where $F \subseteq S1$, and $|S1| \leq |S2| \leq |S_1| + 1$, if possible. In the running example of the official guide we have $S = \{1, 2, \dots, 11\}$ and $S1 = \{1, 2, \dots, 5\}$, $S2 = \{6, 7, \dots, 11\}$.

Eligibility criteria

As described in points C1-C3 in the Appendix, some players are not eligible to be paired. Essentially no two players can be matched twice, no player can become unmatched and thus get a bye twice, and the absolute color preference of a non-topscorer player must be obeyed, so we can never match two non-topscorer players with the same absolute color preference.

Priorities for selecting the pairing

The new version of the FIDE Dutch system rules includes a clear prioritisation order over the pairing selected, which is included in the Appendix.

First, we have to note that slightly different rules are applied for score groups at the end of the process. The so-called completion criteria C4 requires that in the score group before the last one we shall choose the downfloaters in such a way that the last group will admit a complete matching.

The further criteria (C5-C19) are called quality criteria, and indeed these provide the sequential goals that the best pairing should satisfy, see them in the Appendix. The first criterion (C5) is to maximise the size of the pairing within the score group considered. The second criterion (C6) is to maximise the number of downfloaters paired, and among them match the ones with the highest scores. The following rule C7, was not present in the previous version of the Dutch rule, and it is a forward looking rule that requires the selection of the downfloaters in the considered score group such that in the subsequent score group the pairing has maximum size and matches the most downfloaters. The remaining rules C8-C19 provides a specific order how the colour preferences and the repetition of the downfloter and upfloater selection are considered. (Note that here do not consider the exceptions that apply for the very last score group, the first and the last rounds of the tournament, or in case of some other unusual events, e.g. withdrawal or addition of players, unfinished games.)

Tie-breaking by transpositions and exchanges

When the above described priority rules do not provide a unique solution (which is typically the case at the beginning of the tournaments, where the score groups are large and the eligibility and priority criteria are easier to satisfy) the rule suggest a particular order among the possible matchings. The transposition order describes the rankings of the matchings when the set of players to be matched, $S1$ and $S2$, are already fixed. The exchange rules describe in which order one shall try to exchange the players among $S1$ and $S2$ in when the satisfaction level of the priority criteria is already fixed. Thus, in fact the exchange order is more important and we should consider that first, but below we follow the description of the FIDE Handbook and we start describing the transposition orders.

Transposition orders. The most inner process of the Dutch pairing algorithm will select the first feasible pairing between $S1$ and $S2$ as follows. The pairings are sorted according to a lexicographic order considering the first player in $S1$ first, then the second player in $S1$, and so on. For our running example, the players in $S1$ in their order shall get the following opponents, the first suitable pairing from the list described in Table 1.

Exchange orders. If neither of these pairings is eligible then we need to try to exchange players between sets $S1$ and $S2$ in a predefined order, as described in part D2-D3 in the Appendix. For instance, if we can find a suitable pairing by exchanging one pair of players then we should check the pairs to be exchanged in the order described in Table 2.

If more than one pair of players are needed to be exchanged then the rule requires to

1. minimise the number of players exchanged
2. minimise the index differences between the players exchanged
3. lexicographically maximise the indices of the players moved from $S1$ to $S2$

0.	6-7-8-9-10-11
1.	6-7-8-9-11-10
2.	6-7-8-10-11-9
3.	6-7-8-11-9-10
4.	6-7-8-11-9-10
5.	6-7-9-8-10-11
...	...
12.	6-7-10-8-9-11
...	...
24.	6-8-7-9-10-11
...	...
719.	11-10-9-8-7-6

Table 1: Transposition order when pairing $S1 = \{1, 2, 3, 4, 5\}$ and $S2 = \{6, 7, 8, 9, 10, 11\}$.

	5	**4**	**3**	**2**	**1**
6	1	3	6	10	15
7	2	5	9	14	20
8	4	8	13	19	24
9	7	12	18	23	27
10	11	17	22	26	29
11	16	21	25	28	30

Table 2: Priority order when exchanging one pair of players between $S1 = \{1, 2, 3, 4, 5\}$ and $S2 = \{6, 7, 8, 9, 10, 11\}$.

4. lexicographically minimise the indices of the players moved from $S2$ to $S1$

For instance, when considering the exchange of two players from each group, we shall use the following priority order described in Table 3.

3.2 Translating the rules into efficient algorithms

In this section we describe how to translate the selection rules into a maximum weight matching algorithm. Note that here, we focus on the regular cases.

Eligibility requirements. The eligibility requirements (C1-C3) can be easily satisfied by not having edges between the nodes representing these agents in the eligibility graph $G_S(N, E)$ for score group S.

Completion criterion. The completion criterion (C4) is only applied for the score group that is considered before the last one, and as a first priority we have to make it sure that the last score group will have a complete matching. So essentially we have to find a complete matching for the last two score groups. (It is not mentioned in the latest version of the rule what would happen if there exist no complete matching for the last two groups, but in the earlier version they recommend to enlarge the set of players considered with the previous score group(s).)

Quality criteria. Each of the quality criteria (C5-C17) can be translated into a maximum weight matching problem with weight-vectors. For each selection criterion we define a new index for the weight-vector w_e for every edge $e = ij$ as follows.

1. For (C5) we simply set weight 1 for each edge. Maximising this index will ensure that the matching

	5,4	**5,3**	**5,2**	**5,1**	**4,3**	**4,2**	**4,1**	**3,2**	**3,1**	**2,1**
6,7	1	3	7	14	8	16	28	29	45	65
6,8	2	6	13	24	15	27	43	44	64	85
6,9	4	11	22	37	25	41	60	62	83	104
6,10	9	20	35	53	39	58	79	81	102	120
6,11	17	32	50	71	55	76	96	99	117	132
7,8	5	12	23	38	26	42	61	63	84	105
7,9	10	21	36	54	40	59	80	82	103	121
7,10	18	33	51	72	56	77	97	100	118	133
7,11	30	48	69	90	74	94	113	115	130	141
8,9	19	34	52	73	57	78	98	101	119	134
8,10,	31	49	70	91	75	95	114	116	131	142
8,11	46	67	88	108	92	111	126	128	139	146
9,10	47	68	89	109	93	112	127	129	140	147
9,11	66	87	107	123	110	125	137	138	145	149
10,11	86	106	122	135	124	136	143	144	148	150

Table 3: Priority order when exchanging two pairs of players between $S1 = \{1, 2, 3, 4, 5\}$ and $S2 = \{6, 7, 8, 9, 10, 11\}$, as given in the FIDE Handbook.

in maximum size.

2. For (C6) we set weight 1 if either i or j is a downfloater, and 0 otherwise. (Note that there cannot be an edge between two downfloaters, since in that we would had matched them before). This weighting will ensure that we match as many downfloaters as possible.

3. Still corresponding to (C6), we need to match first those downfoaters who have the highest scores and continue with the second highest ones. This can be achieved by adding a weight-vector for every eligible pair containing a downfloater, which is a zero-one vector as long, as the number of different scores of the downfloaters. For instance, if the score group considered contains players with score 4 and there are downfloaters with scores 5.5 and 4.5 then we add a vector of length two, and first we put a value 1 to those players with score 5.5 and then a value 1 for those with score 4.5, leaving the other values zero.

4. Rule (C7) is a special one, as we will need to ensure the maximality of the matching in the subsequent score group, denoted by S', and also the number of downfloaters matched there. For this rule, we extend graph G_S to graph $G_{S \cup S'}$, and we only define weights with regard to this index for edges between S and S' and within S'. Let the weight of these edges be 1, ensuring first the maximality of the matching in the subsequent score group.

5. To ensure that the number of downfloaters matched is also maximal when matching the subsequent group S', according to (C7), we add weight 1 of each edge between S and S'.

6. Rules (C8)-(C9) only apply for the topscorers and their opponents in the last round (i.e. "players who have a score of over 50% of the maximum possible score when pairing the final round of the tournament"), as a relaxation of eligibility criteria (C3). In case we are considering these players we add weight -1 for those edges where both players have the same absolute colour preference, first by the fact that either they both have +2 or -2 colour difference.

7. Continuing the above rule by part (C9), we also add weight -1 for those pairs who both played with the same colour two times in a row.

8. To ensure that most player get their colour preferences according to (C10), we add weight -1 if both players have the same colour preference.

9. Similarly, we can minimise the number of player who do not get their strong colour preference, as required in (C11), by adding weight -1 if both players have the same strong colour preference.

10. To satisfy (C12), if either of the two players involved was a downfloater in the previous round then we add weight 1, so the algorithm will try to match as many of them as possible, and avoid to select them to become downfloaters again.

11. Minimising the selections of the same players for becoming upfloaters, as described in (C13), we add weight -1 for edge ij if i is a current downfloater (i.e. i was unmatched in the previous score group), and j was an upfloater in the previous round.

12. Selection rule (C14) can be treated in the same way as rule (C12).

13. Selection rule (C15) can be treated in the same way as rule (C13).

14. In rule (C16) we need to minimise the score differences for the players who receive the same downfloat as in the previous round. So, if for pair ij, $i \in S$ was a downfloater in the previous round and $j \in S'$ then we add $-k$ as the weight if k is the difference between the scores of player i and j.

15. Similarly, in rule (C17) we minimise the score differences from the point of view of the repeated upfloaters, by adding weight $-k$ if the difference between the scores of downfloater i and previous upfloater j is k.

16. Rule (C18) can be treated as rule (C16).

17. Rule (C19) can be treated as rule (C17).

Finding a lexicographically weight-maximal matching with the above weighting on $G_{S \cup S'}$ will provide us a matching that we would select when sequentially maximising criteria (C5-C19). After optimising with regard to the quality criteria, we need to choose the pairing according to the transposition and exchange rules. Since the exchange rules are superior, we start the translation with that.

Exchange rule. We extend the above weight-vectors with the following components, responsible for enforcing the exchange selection. Here we describe the translation for so-called homogenous score groups (where no downfloaters are present), but the heterogenous case can be treated similarly.

1. To minimise the number of players exchanged we add weights -1 for every edge within $S1$ and within $S2$. Our optimal matching will use as few edges as possible, which also means that the number of players exchanged is minimal.

2. To minimise the index differences between the players exchanged we add the following negative weights. Let r_i be the index of player i, and let a_S denote the index between the highest index in $S1$ and the lowest index in $S2$ (this is 5.5 in our running example). For every edge ij, where $i, j \in S1$ and $r_i < r_j$ let the weight of ij in the vector be $r_j - a_S$. Similarly, for every edge ij, where $i, j \in S2$ and $r_i < r_j$ let the weight of ij in the vector be $a_S - r_i$. E.g. for edge $\{2, 4\}$ in our running example this weight is -1.5 and for edge $\{8, 9\}$ this weight is -2.5.

3. To lexicographically maximise the indices of the players moved from $S1$ to $S2$, we add a weight-vector of length $|S1|$ and with one nonzero element, as follows. If $i, j \in S1$ with $r_i < r_j$ then we add a weight 1 to the $\lceil a_S - r_j \rceil$-th coordinate. For instance, in our running example when considering edge $\{2, 4\}$, the added weight-vector is $[0, 1, 0, 0, 0]$. This weighting will ensure that we will move player 5 to $S1$ whenever it is possible, and if not then player 4, and so on.

4. To lexicographically minimise the indices of the players moved from $S2$ to $S1$ we further extend the weight-vector with a new component of length $|S2|$ and with one nonzero element. If $i, j \in S2$ with $r_i < r_j$ then we add a weight 1 to the $\lceil r_i - a_S \rceil$-th coordinate. For instance, when considering edge $\{8, 9\}$ in the running example, the added vector is $[0, 0, 1, 0, 0, 0]$. Thus we move player 6 first, if possible, then player 7, and so on.

Finally, for choosing the first pairing among the so far optimal ones, we translate the selection according to the transposition order into a maximum weight matching problem.

Transposition rule. With the transposition rule, we assume that the partition $S1 \cup S2$ is already fixed and we would like to ensure that the among the possible pairings we first select the partner of the player with the smallest index in $S1$ to be the player with the smallest index in $S2$, and if this is not possible then the player with the second smallest index in $S2$, and so on. After selecting the partner of the highest ranked player in $S1$, we continue with selecting a partner for the second highest ranked player in $S1$, and so on. To achieve this, we add another weight-vector component to each edge of length $|S| - 1$ as follows. For ij, where $r_i < r_j$ we add weight $-r_j$ on the r_i-th position in this vector and we keep the other position zero-valued. For instance, for edge $\{2, 7\}$ in our example, we add vector $[0, -7, 0, 0, 0, 0, 0, 0, 0, 0]$.

4 Further notes

In this paper we discussed how to replace the priority rules in the Swiss pairing systems by efficient algorithms. However, we have only studied the Dutch variant, and we have not considered some special cases (last round, heterogenous score groups, new or leaving players, etc). Nevertheless, we believe that all of the variants of the Swiss pairings can be completely conducted by efficient algorithms, so our first future plan is to investigate the remaining details of the Dutch rule and the three other systems.

If we succeed to translate the official pairing procedures into sophisticated efficient algorithms then we can incorporate them into a software and conduct further studies. In particular, it would be interesting to simulate tournaments and compare the performance of the four variants with respect to their success of ranking the players according to their real strength within the same number of rounds. This would follow up the research of Csató [1], who compared the final rankings of some particular tournaments organised by Swiss pairings with other ranking methods.

In a future research one could also investigate the performance of these variants from a more general point of view, by considering the utilities of the players. A player in a Swiss tournament may not be really interested in her final ranking, and the ranking of the others, as perhaps she just wishes to play with opponents of similar strength. Note that such preferences are not likely to be satisfied in the most widely used Dutch variant if a tournament has many participants, since according to the exhaustive search procedure (dividing a score group based to the ELO point of the players and then trying to match them according to their order in their subgroups), a typical player will either play with much stronger of with much weaker players in the first 5-6 rounds of the 9-round tournament. One alternative pairing method would consider the preferences of the players and match them e.g. with a stable matching algorithm, as proposed in [4]. Thus the four variants could be compared with respect to such preferences, namely how large is the gap between the strengths of the paired players in average during a tournament. Finally, it would also be interesting to see what kind of alternative pairings could be used to better satisfy the preferences of the players when the classical goal of selecting a winner and ranking the others is ignored.

References

[1] L. Csató. Ranking in Swiss system chess team tournaments. *Corvinus Economics Working Papers (CEWP) 2015/01, Corvinus University of Budapest*, 2015.

[2] H.N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. *In proceedings of SODA-1990: The first annual ACM-SIAM symposium on Discrete algorithms*, 1990.

Gabow, Harold N. "Data structures for weighted matching and nearest common ancestors with linking." Society for Industrial and Applied Mathematics, 1990.

[3] World Chess Federation (FIDE) Website. www.fide.com/fide/handbook. *Accessed on 14 February 2016.*

[4] E. Kujansuu, T. Lindberg, E. Mäkinen. The Stable Roommates Problem and Chess Tournament Pairings. *Divulgaciones Matemáticas*, 7(1):19–28, 1999.

[5] S. Ólafsson. Weighted Matching in Chess Tournaments. *The Journal of the Operational Research Society*, 41(1):17–24, 1990.

[6] S. Micali, V.V. Vazirani. An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs. *In: Proceedings of FOCS 1980: the 21st Annual Symposium on Foundations of Computer Science*, 17–27, 1980.

Appendix

FIDE Handbook, the Dutch system (to be applied from 1 July 2017)

C Pairing Criteria

Absolute Criteria

No pairing shall violate the following absolute criteria:
C.1 see C.04.1.b (Two players shall not play against each other more than once)
C.2 see C.04.1.d (A player who has already received a pairing-allocated bye, or has already scored a (forfeit) win due to an opponent not appearing in time, shall not receive the pairing-allocated bye).
C.3 non-topscorers (see A.7) with the same absolute colour preference (see A6.a) shall not meet (see C.04.1.f and C.04.1.g).

Completion Criterion
C.4 if the current bracket is the PPB (see A.9): choose the set of downfloaters in order to complete the roundpairing.

Quality Criteria
To obtain the best possible pairing for a bracket, comply as much as possible with the following criteria, given in descending priority:
C.5 maximize the number of pairs (equivalent to: minimize the number of downfloaters).
C.6 minimize the PSD (This basically means: maximize the number of paired MDP(s); and, as far as possible, pair the ones with the highest scores).
C.7 if the current bracket is neither the PPB nor the CLB (see A.9): choose the set of downfloaters in order first to maximize the number of pairs and then to minimize the PSD (see C.5 and C.6) in the following bracket (just in the following bracket).
C.8 minimize the number of topscorers or topscorers' opponents who get a colour difference higher than +2 or lower than -2.
C.9 minimize the number of topscorers or topscorers' opponents who get the same colour three times in a row.
C.10 minimize the number of players who do not get their colour preference.
C.11 minimize the number of players who do not get their strong colour preference.
C.12 minimize the number of players who receive the same downfloat as the previous round.
C.13 minimize the number of players who receive the same upfloat as the previous round.
C.14 minimize the number of players who receive the same downfloat as two rounds before.
C.15 minimize the number of players who receive the same upfloat as two rounds before.

C.16 minimize the score differences of players who receive the same downfloat as the previous round.
C.17 minimize the score differences of players who receive the same upfloat as the previous round.
C.18 minimize the score differences of players who receive the same downfloat as two rounds before.
C.19 minimize the score differences of players who receive the same upfloat as two rounds before.

D Rules for the sequential generation of the pairings

Before any transposition or exchange take place, all players in the bracket shall be tagged with consecutive in-bracket sequence-numbers (BSN for short) representing their respective ranking order (according to A.2) in the bracket (i.e. 1, 2, 3, 4, ...).

D.1 Transpositions in $S2$

A transposition is a change in the order of the BSNs (all representing resident players) in $S2$.
All the possible transpositions are sorted depending on the lexicographic value of their first N1 BSN(s), where N1 is the number of BSN(s) in $S1$ (the remaining BSN(s) of $S2$ are ignored in this context, because they represent players bound to constitute the remainder in case of a heterogeneous bracket; or bound to downfloat in case of a homogeneous bracket - e.g. in a 11-player homogeneous bracket, it is 6-7-8-9-10, 6-7-8-9-11, 6-7-8-10-11, ..., 6-11-10-9-8, 7-6-8-9-10, ..., 11-10-9-8-7 (720 transpositions); if the bracket is heterogeneous with two MDPs, it is: 3-4, 3-5, 3-6, ..., 3-11, 4-3, 4-5, ..., 11-10 (72 transpositions)).

D.2 Exchanges in homogeneous brackets or remainders (original $S1 \iff$ original $S2$)

An exchange in a homogeneous brackets (also called a resident-exchange) is a swap of two equally sized groups of BSN(s) (all representing resident players) between the original $S1$ and the original $S2$. In order to sort all the possible resident-exchanges, apply the following comparison rules between two resident-exchanges in the specified order (i.e. if a rule does not discriminate between two exchanges, move to the next one).
The priority goes to the exchange having:
a) the smallest number of exchanged BSN(s) (e.g exchanging just one BSN is better than exchanging two of them).
b) the smallest difference between the sum of the BSN(s) moved from the original $S2$ to $S1$ and the sum of the BSN(s) moved from the original $S1$ to $S2$ (e.g. in a bracket containing eleven players, exchanging 6 with 4 is better than exchanging 8 with 5; similarly exchanging 8+6 with 4+3 is better than exchanging 9+8 with 5+4; and so on).
c) the highest different BSN among those moved from the original $S1$ to $S2$ (e.g. moving 5 from $S1$ to S2 is better than moving 4; similarly, 5-2 is better than 4-3; 5-4-1 is better than 5-3-2; and so on).
d) the lowest different BSN among those moved from the original $S2$ to $S1$ (e.g. moving 6 from $S2$ to S1 is better than moving 7; similarly, 6-9 is better than 7-8; 6-7-10 is better than 6-8-9; and so on).

D.3 Exchanges in heterogeneous brackets (original $S1 \iff$ original Limbo) An exchange in a heterogeneous bracket (also called a MDP-exchange) is a swap of two equally sized groups of BSN(s) (all representing MDP(s)) between the original $S1$ and the original Limbo. In order to sort all the possible MDP-exchanges, apply the following comparison rules between two MDP-exchanges in the specified order (i.e. if a rule does not discriminate between two exchanges, move to the next one) to the players that are in the new $S1$ after the exchange.
The priority goes to the exchange that yields a $S1$ having:
a) the highest different score among the players represented by their BSN (this comes automatically in complying with the C.6 criterion, which says to minimize the PSD of a bracket).
b) the lowest lexicographic value of the BSN(s) (sorted in ascending order).
Any time a sorting has been established, any application of the corresponding D.1, D.2 or D.3 rule, will pick the next element in the sorting order.

Spanning trees and logarithmic least squares optimality for complete and incomplete pairwise comparison matrices

Sándor Bozóki[1]

Institute for Computer Science and Control,
Hungarian Academy of Sciences (MTA SZTAKI)
13-17 Kende str., Budapest, 1111, Hungary

Corvinus University of Budapest
bozoki.sandor@sztaki.mta.hu

Vitaliy Tsyganok

Laboratory for Decision Support Systems
The Institute for Information Recording of
National Academy of Sciences of Ukraine
2, Shpak str., Kyiv, 03113, Ukraine

Department of System Analysis
State University of Telecommunications, Ukraine
tsyganok@ipri.kiev.ua

Abstract: Pairwise comparison matrices provide a user-friendly way of cardinal preference modelling. Decision makers compare the importance of criteria, or the performance of alternatives with respect to a given criterion. Numerical answers are arranged into a square matrix, which is element-wise reciprocal of its own transpose. A pairwise comparison matrix can be completely filled in (complete) or incomplete. Incomplete pairwise comparison matrices offer a wider range of applicability, not only in multi-criteria decision making, but in ranking problems as well.

The objective is to determine weights that express the importance of criteria, or the scores of the alternatives with respect to a criterion, by numbers, such that the pairwise ratios of the weights are as close as possible to the matrix elements, given by the decision maker. Several distance minimizing methods have been proposed, as well as other methods without the specification of the metric. The spanning tree approach belongs to the second group by definition. However, Lundy, Siraj and Greco recently proved that the geometric mean of the weight vectors, calculated from all spanning trees of a complete pairwise comparison matrix's graph, is in fact the optimal solution of the logarithmic least squares problem.

We generalize this result for the class of incomplete pairwise comparison matrices.

Keywords: multi-criteria decision making, pairwise comparison matrix, spanning tree, logarithmic least squares

1 Introduction

Definition 1 *Let $n \geq 2$ be an integer. Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called a* pairwise comparison matrix*, if $a_{ij} > 0$ and $a_{ij} = 1/a_{ji}$ for all $1 \leq i, j \leq n$.*

Remark 2 *The element-wise logarithm of a pairwise comparison matrix is a skew-symmetric matrix.*

Definition 3 *A pairwise comparison matrix is called* incomplete *if some of its elements are missing.*

Definition 4 *Given an (in)complete pairwise comparison matrix $\mathbf{A}$ of size $n \times n$, its* associated undirected graph *is defined as follows: it has n nodes and the edge between nodes i and j is drawn if and only if the matrix element a_{ij} is known.*

[1]The support of the János Bolyai Research Fellowship of the Hungarian Academy of Sciences no. BO/00154/16/3 and the Hungarian Scientific Research Fund (OTKA) grant K111797 is greatly acknowledged.

Definition 5 *The* logarithmic least squares problem *is as follows:*

$$\min \sum_{\substack{i,\,j\,:\\ a_{ij}\,is\ known}} \left[\log a_{ij} - \log\left(\frac{w_i}{w_j}\right)\right]^2$$
$$w_i > 0, \qquad i = 1, 2, \ldots, n,$$
$$\sum_{i=1}^{n} w_i = 1.$$

Normalization constraint is technical, and it is often replaced by $\prod_{i=1}^{n} w_i = 1$ or by $w_1 = 1$.

Every spanning tree of the graph associated to the (in)complete pairwise comparison matrix induces a unique, up to scalar multiplication, weight vector.

Theorem 6 *(Lundy, Siraj and Greco [2]) The geometric mean of weight vectors calculated from all spanning trees is logarithmic least squares optimal in case of complete pairwise comparison matrices.*

2 Main result

Theorem 7 *[1] Let* **A** *be an incomplete or complete pairwise comparison matrix such that its associated graph is connected. Then the optimal solution to the logarithmic least squares problem is equal, up to a scalar multiplier, to the geometric mean of weight vectors calculated from all spanning trees.*

Proof: The proof can be found in [1]. □

Main references

[1] Bozóki, S., and Tsyganok, V. The logarithmic least squares optimality of the geometric mean of weight vectors calculated from all spanning trees for (in)complete pairwise comparison matrices, *Under review,* https://arxiv.org/abs/1701.04265 (2017)

[2] Lundy, M., Siraj, S., and Greco, S. The mathematical equivalence of the "spanning tree" and row geometric mean preference vectors and its implications for preference analysis. *European Journal of Operational Research* **257**(1):197–208 (2017)

Two Extensions of a Theorem of Tutte

Gunnar Brinkmann

Department of Applied Mathematics,
Computer Science and Statistics
Ghent University
Krijgslaan 281-S9, 9000 Ghent, Belgium
Gunnar.Brinkmann@UGent.be

Kenta Ozeki[1]

Graduate School of Environment and
Information Sciences
Yokohama National University
79-7 Tokiwadai, Hodogaya-ku,
Yokohama 240-8501, Japan
ozeki-kenta-xr@ynu.ac.jp

Carol T. Zamfirescu[2]

Department of Applied Mathematics,
Computer Science and Statistics
Ghent University
Krijgslaan 281-S9, 9000 Ghent, Belgium
czamfirescu@gmail.com

Abstract: In this talk, we present two extensions of Tutte's famous theorem stating that 4-connected planar graphs are hamiltonian. We show that (i) planar 3-connected graphs with at most three 3-vertex-cuts are hamiltonian, and that (ii) every 4-connected graph with crossing number at most 2 is hamiltonian. (i) is based on joint work with Gunnar Brinkmann, while (ii) was obtained together with Kenta Ozeki.

Keywords: Polyhedral graph, hamiltonian, 3-cut, crossing number.

1 Polyhedra with at most three 3-cuts

In 1931, Whitney showed that 4-connected triangulations of the plane are hamiltonian [10]. This result was generalised by Tutte in 1956, who showed that in fact every planar 4-connected graph is hamiltonian [9]. In this talk, we strengthen this classic theorem in two ways.

In the following, a *polyhedron* is a planar 3-connected graph, all cuts shall be vertex-cuts, and a cut of cardinality k will be called a *k-cut*. Let G be a polyhedron and $X = \{u, v, w\}$ a 3-cut in G. If (V', E') is a component of $G - X$, then $G[V' \cup X]$ is called a *closed component* of $G - X$. If (V'', E'') is a closed component of $G - X$, then $(V'', E'' \cup \{uv, vw, wu\})$ is called an *edge closed component* of $G - X$. We shall outline the main ideas behind these notions, linking closed components and edge closed components to our pursuit of hamiltonicity, and emphasise the most important difference between planar triangulations and polyhedra with respect to their 3-cuts and their hamiltonian properties.

Various stronger versions of Whitney's theorem have appeared, with one of the most far-reaching results being a theorem of Jackson and Yu [3] stating that even if we allow up to three separating triangles to occur in a plane triangulation, hamiltonicity is still guaranteed. Tutte's seminal theorem has been generalised in several ways as well—see for instance Sanders' result that in a 4-connected plane graph a hamiltonian cycle through any two edges exists [7]. However, the Jackson-Yu theorem was not generalised to all polyhedra with at most three 3-cuts. Gunnar Brinkmann and I have done so in [1]. The first part of the talk will revolve around this result. Let us emphasise that the theorem of Jackson and Yu

[1]This work was partially supported by JSPS KAKENHI Grant Number 25871053.
[2]This research is supported by a Postdoctoral Fellowship of the Research Foundation Flanders (FWO).

does not only concern the number of 3-cuts, but also their relative position, encoded by a *decomposition tree*. Such decomposition trees, which are unique for triangulations, are not defined for general plane graphs, so only the part of the Jackson-Yu result concerning the number of 3-cuts can be generalised.

A graph G is *k-hamiltonian* if for each set $S \subset V(G)$ of cardinality k, the graph $G - S$ is hamiltonian. In 1994, Thomas and Yu proved the following result which was originally conjectured by Plummer.

Theorem 1 (Thomas and Yu [8]) *4-connected polyhedra are 2-hamiltonian.*

We present the following useful lemma.

Lemma 2 (Brinkmann and Zamfirescu [1]) *A polyhedron G with k 3-cuts contains a spanning subgraph that can be obtained from a 4-connected polyhedron by deleting at most k vertices.*

Together with this lemma, Theorem 1 implies:

- Polyhedra with at most two 3-cuts are hamiltonian.
- Polyhedra with at most one 3-cut are 1-hamiltonian.

It is clear that Theorem 1 cannot be strengthened to imply 3-hamiltonicity, so in order to prove that polyhedra with at most three 3-cuts are hamiltonian, we need a different strategy. In the talk we will present an important ingredient of the proof of our main theorem from [1], which now follows.

Theorem 3 (Brinkmann and Zamfirescu [1]) *Polyhedra with at most three 3-cuts are hamiltonian.*

We also proved that polyhedra with at most four 3-cuts are traceable (i.e. contain a hamiltonian path). A natural question is how few 3-cuts a non-hamiltonian or non-traceable polyhedron may have. We do not have a definitive answer, but the following is known.

Proposition 4 (Brinkmann, Souffriau, and Van Cleemput [2]) *For all $k \geq 6$ there exist non-hamiltonian triangulations with exactly k 3-cuts.*

Proposition 5 (Brinkmann and Zamfirescu [1]) *For all $k \geq 8$ there exist non-traceable triangulations with exactly k 3-cuts.*

Combining the above conclusions, the obvious open questions are:

- Are polyhedra with four or five 3-cuts hamiltonian?
- Are polyhedra with five, six or seven 3-cuts traceable?

If time permits, we will give an overview of what is known concerning other hamiltonian properties (such as hamiltonian-connectedness) in polyhedra with few 3-cuts [5].

2 4-connected graphs with crossing number 2

In the second part of the talk, we see "planar" as "having crossing number 0". What happens with hamiltonicity in 4-connected graphs with non-vanishing crossing number? Since every graph with crossing number 1 is projective-planar, and making use of the result of Kawarabayashi and Ozeki [4] that every 4-connected projective-planar graph is hamiltonian-connected, we have that every 4-connected graph with crossing number 1 is hamiltonian-connected. Another useful result is due to Brinkmann (see [11]): by using Theorem 1, he showed that if e and f are the crossing edges in a 4-connected graph G with crossing number 1, then there exists a hamiltonian cycle in $G - \{e, f\}$.

Making use of this theorem of Brinkmann and results of Thomas and Yu [8], as well as a lemma of Sanders [7], we were able to prove the following.

Theorem 6 (Ozeki and Zamfirescu [6]) *Every* 4*-connected graph with crossing number* 2 *is hamiltonian.*

We also showed:

Proposition 7 (Ozeki and Zamfirescu [6]) *Every* 4*-connected graph with crossing number at most* 5 *is* 1*-tough.*

For a graph G, the number of 3-cuts shall be denoted with $\phi(G)$, and its crossing number with $\mathrm{cr}(G)$. We link these numbers and hamiltonicity in the following way:

Proposition 8 (Ozeki and Zamfirescu [6]) *For every pair of non-negative integers* k, ℓ *with* $k+\ell = 6$ *(*k', ℓ' *with* $k' + \ell' = 8$*), there exist infinitely many non-hamiltonian (non-traceable)* 3*-connected graphs* G *with* $\phi(G) = k$ *and* $\mathrm{cr}(G) = \ell$ *(*$\phi(G) = k'$ *and* $\mathrm{cr}(G) = \ell'$*).*

References

[1] G. Brinkmann and C. T. Zamfirescu, A Strengthening of a Theorem of Tutte on Hamiltonicity of Polyhedra, Submitted.

[2] G. Brinkmann, J. Souffriau, and N. Van Cleemput, On the strongest form of a theorem of Whitney for hamiltonian cycles in plane triangulations, *J. Graph Theory* **83** (2016) 78–91.

[3] B. Jackson and X. Yu, Hamilton cycles in plane triangulations, *J. Graph Theory* **41** (2002) 138–150.

[4] K. Kawarabayashi and K. Ozeki, 4-connected projective-planar graphs are Hamiltonian-connected, *J. Combin. Theory Ser. B* **112** (2015) 36–69.

[5] K. Ozeki, N. Van Cleemput, and C. T. Zamfirescu, Hamiltonian properties of polyhedra with few 3-cuts – A survey, Submitted.

[6] K. Ozeki and C. T. Zamfirescu, 4-connected graphs with crossing number 2 are hamiltonian, In preparation.

[7] D. P. Sanders, On paths in planar graphs, *J. Graph Theory* **24** (1997) 341–345.

[8] R. Thomas and X. Yu, 4-connected projective-planar graphs are hamiltonian, *J. Combin. Theory, Ser. B* **62** (1994) 114–132.

[9] W. T. Tutte, A theorem on planar graphs, *Trans. Amer. Math. Soc.* **82** (1956) 99–116.

[10] H. Whitney, A theorem on graphs, *Ann. Math.* **32** (1931) 378–390.

[11] C. T. Zamfirescu, Cubic vertices in planar hypohamiltonian graphs, Submitted.

Characterizing brace-minimal rigidity of square-grid frameworks with holes

Siu-Wing Cheng

Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China
scheng@cse.ust.hk

Yuya Higashikawa

Department of Information and System Engineering, Chuo University, Japan
CREST, Japan Science and Technology Agency (JST), Japan
higashikawa.874@g.chuo-u.ac.jp

Naoki Katoh

Department of Informatics, Kwansei Gakuin University, Japan
CREST, Japan Science and Technology Agency (JST), Japan
naoki.katoh@gmail.com

Adnan Sljoka

Department of Informatics, Kwansei Gakuin University, Japan
CREST, Japan Science and Technology Agency (JST), Japan
adnanslj@gmail.com

Abstract: The *rigidity* of *square-grid frameworks* was first studied by Bolker and Crapo [1]. They gave a combinatorial characterization for the rigidity of frameworks with no holes. After two decades, Gáspár, Radics and Recski [2] extended the result to frameworks with holes and provided a method to determine the rigidity faster than computing the rank of its *rigidity matrix*. While the characterization by Bolker et al. immediately provides a necessary and sufficient condition for the *brace-minimal rigidity* of frameworks with no holes, one by Gáspár et al. does not provide such an explicit condition for the case with holes. In this paper, we give the first necessary and sufficient condition for the brace-minimal rigidity of square-grid frameworks with holes.

Keywords: Combinatorial rigidity; Bar-joint framework; Square-grid framework; Bracing

1 Introduction

A *d-dimensional bar-joint framework* is a collection of one-dimensional rigid *bars* connected by zero-dimensional *joints* in $\mathbb{R}^d$. Each pair of bars connected by a joint is allowed to move continuously so that its relative motion is a d-dimensional rotation around the joint. For each joint, we define its *infinitesimal motion* as a d-dimensional velocity vector. Each bar imposes a linear constraint on the infinitesimal motions of its two end joints (for more detail, see [7]). We thus have a homogeneous system of $\{\#$ bars$\}$ linear equations with $d \times \{\#$ joints$\}$ variables. The framework is called *infinitesimally rigid* if such a system admits the D-dimensional solution space, where D is the degree of freedom of a rigid body in $\mathbb{R}^d$, i.e., $\binom{d+1}{2}$. Here the coefficient matrix of such a system is called the *rigidity matrix*, and a framework is infinitesimally rigid if the rank of its rigidity matrix is $d \times \{\#$ joints$\} - D$. This implies that at least $d \times \{\#$ joints$\} - D$ bars are necessary for the infinitesimal rigidity of a framework, which was first formulated by Maxwell [5].

Laman [4] established the necessary and sufficient combinatorial characterization for the infinitesimal rigidity of *generic* two-dimensional bar-joint frameworks. A framework is called generic if the rank of its rigidity matrix and its row-induced submatrices take the maximum values over all frameworks which are

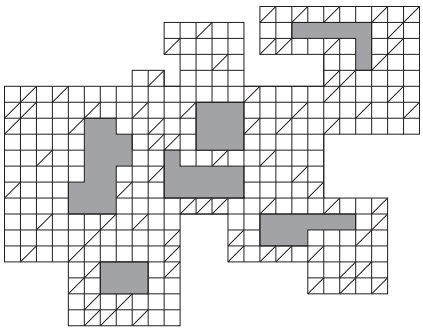


Figure 1: A square-grid framework with holes

the same topologically (also see [7]). However, combinatorial characterization for the case when $d \geq 3$ has not been found yet.

Even in two-dimension, where there are many applications in the real world (e.g., civil engineering or architectural engineering), it is worth to study non-generic frameworks. Based on this, we study *square-grid frameworks* in this paper. A square-grid framework is a connected two-dimensional bar-joint framework consisting of a union of unit grid squares, where some contain diagonal braces (see Figure 1). We assume that (i) all joints are located at the integer grid, (ii) the area of every hole is at least two, and (iii) any two of the outer face and holes of a framework do not share any joints.

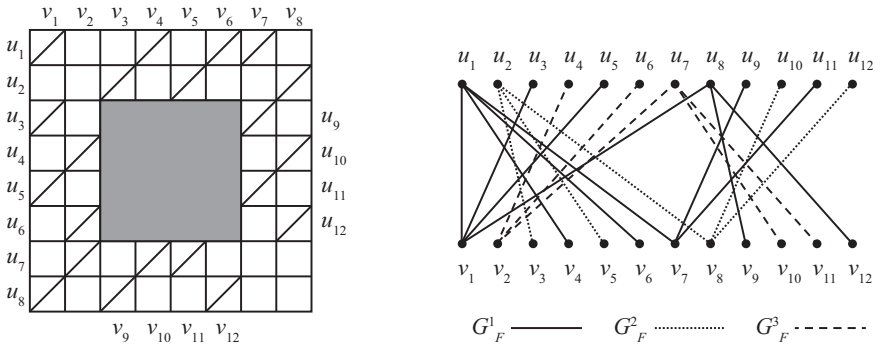
In the literature, the rigidity of square-grid frameworks was first studied by Bolker and Crapo [1]. They gave a combinatorial characterization for the infinitesimal rigidity of frameworks with no holes (which will be shown at Theorem 1). Also Gáspár, Radics and Recski [2] studied the case with holes and provided a method to determine the infinitesimal rigidity of a framework, which is faster than computing the rank of its rigidity matrix (see Theorem 2). Ito, Kobayashi, Higashikawa, Katoh, Poon and Saumell [3] have recently proposed an algorithm for the bracing problem: given a square-grid framework with holes in which there is no brace, the objective is to add the minimum number of braces which makes the framework infinitesimally rigid.

A square-grid framework is called *brace-minimally rigid* if the framework is infinitesimally rigid and removing any brace makes the framework infinitesimally flexible. Then the characterization by [1] immediately provides a necessary and sufficient condition for the brace-minimal rigidity of a framework with no holes, however the results by [2, 3] do not provide such an explicit condition for the case with holes (though the result by [2] provides a brute-force way to check the brace-minimal rigidity of a square-grid framework with holes, see just after Theorem 3). In this paper, we give the first necessary and sufficient condition for the brace-minimal rigidity of a square-grid framework with holes.

2 Preliminaries

In a square-grid framework, there are two types of bars (other than braces), that is, *horizontal-bars* and *vertical-bars* (for short, h-bars and v-bars). We define an *h-strip* (resp. a *v-strip*) as a maximal set of horizontally (resp. vertically) consecutive grid squares. Let us give all h-strips and v-strips indices, respectively. If the i-th h-strip and the j-th v-strip intersect each other at a unit grid square, we call it square (i, j). For each h-bar (resp. v-bar), we define an *infinitesimal rotation* as the difference between the vertical (resp. horizontal) components of its two end joint's infinitesimal motions. As observed in [1, 2], infinitesimal rotations of all v-bars (resp. h-bars) in an h-strip (resp. a v-strip) are the same, called an infinitesimal rotation of the h-strip (resp. v-strip). In addition, if square (i, j) is braced, infinitesimal rotations of the i-th h-strip and the j-th v-strip are the same. In the subsequent discussion, we use "rigid" and "rotation" to denote "infinitesimally rigid" and "infinitesimal rotation", respectively.

Given a square-grid framework F, let G_F denote a bipartite graph consisting of the vertex sets U_F, V_F and the edge set E_F such that $u_i \in U_F$ corresponds to the i-th h-strip in F, $v_j \in V_F$ corresponds to the



(a) F: a framework with a hole

(b) G_F: a bipartite graph consisting of connected components G^1_F, G^2_F and G^3_F

Figure 2: Illustrations of a square-grid framework F and the corresponding bipartite graph G_F

j-th v-strip in F, and for $u_i \in U_F$ and $v_j \in V_F$ an edge $(u_i, v_j) \in E_F$ if square (i, j) is braced in F (see Figure 2). In the following, if $u \in U_F$ (resp. $v \in V_F$) corresponds to an h-strip (resp. v-strip) in F, we treat u (resp. v) as the h-strip (resp. v-strip) itself. Also, if $e \in E_F$ corresponds to a brace in F, we treat e as the brace itself. Let us observe the theorem by Bolker et al. [1].

Theorem 1 [1] *A square-grid framework F with no holes is rigid if and only if G_F is connected.*

Suppose that there are q connected components of G_F, say $G^1_F, \ldots, G^q_F$ (see Figure 2(b)). For some integer $l \in \{1, \ldots, q\}$, let U^l_F, V^l_F and E^l_F denote subsets of U_F, V_F and E_F such that $G^l_F = (U^l_F, V^l_F, E^l_F)$, respectively. We say that the i-th h-strip (resp. the j-th v-strip) belongs to G^l_F if $u_i \in U^l_F$ (resp. $v_j \in V^l_F$). If $u_i \in U^l_F$, $v_j \in V^l_F$ and square (i, j) is braced in F, we also say that the brace at square (i, j) belongs to G^l_F.

We introduce the *hole matrix* of F, which was first defined in [2]. Suppose that there are p holes in F, say *holes* $1, \ldots, p$. Let us focus on hole $k \in \{1, \ldots, p\}$. We define a *left h-strip* for hole k as an h-strip such that the rightmost v-bar in the h-strip is on the boundary of hole k. Similarly, we also define a *right h-strip*, an *upper v-strip* and a *lower v-strip* for hole k (see Figure 3). Then, as observed in [2], we can see that for any hole $k \in \{1, \ldots, p\}$

$$\sum\{\text{rotations of left h-strips}\} - \sum\{\text{rotations of right h-strips}\} = 0, \quad \text{and} \tag{1}$$

$$\sum\{\text{rotations of upper v-strips}\} - \sum\{\text{rotations of lower v-strips}\} = 0. \tag{2}$$

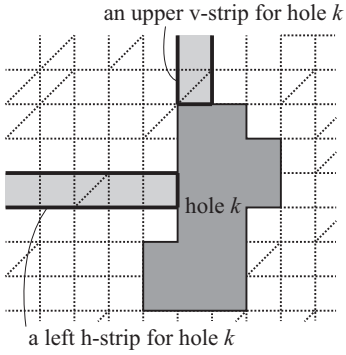


Figure 3: Illustration of a left h-strip and an upper v-strip for hole k

Note that rotations of all h-strips and v-strips belonging to a connected component are the same. Let ω_l denote a rotation of G_F^l. For integers $k \in \{1, \ldots, p\}$ and $l \in \{1, \ldots, q\}$, let α_{kl}^+ (resp. α_{kl}^-, β_{kl}^+ and β_{kl}^-) be the number of left h-strips (resp. right h-strips, upper v-strips and lower h-strips) for hole k belonging to G_F^l. Then, equations (1) and (2) can be written as

$$\sum_{l \in \{1,\ldots,q\}} (\alpha_{kl}^+ - \alpha_{kl}^-)\omega_l = 0 \quad \forall k \in \{1, \ldots, p\}, \quad \text{and} \tag{3}$$

$$\sum_{l \in \{1,\ldots,q\}} (\beta_{kl}^+ - \beta_{kl}^-)\omega_l = 0 \quad \forall k \in \{1, \ldots, p\}. \tag{4}$$

Letting $\alpha_{kl} = \alpha_{kl}^+ - \alpha_{kl}^-$ and $\beta_{kl} = \beta_{kl}^+ - \beta_{kl}^-$, we obtain a system of the above equations as

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1q} \\ \beta_{11} & \beta_{12} & \cdots & \beta_{1q} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{p1} & \beta_{p2} & \cdots & \beta_{pq} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{5}$$

We call the coefficient matrix in (5) the hole matrix of F, denoted by H_F. Let us observe the theorem by Gáspár et al. [2].

Theorem 2 [2] *Given a square-grid framework F with p holes, suppose that G_F consists of q connected components. Then F is rigid if and only if the rank of H_F is $q-1$.*

We now state the main theorem below, which will be proved in Section 3.

Theorem 3 *Given a square-grid framework F with p holes, suppose that G_F consists of q connected components. Then F is brace-minimally rigid if and only if (a) the rank of H_F is $q-1$, (b) G_F is a spanning forest, and (c) $q = 2p+1$.*

Note that applying Theorem 2, we can also determine the brace-minimal rigidity of F as follows: Check the rigidity of F and $F-e$ for every brace $e \in E_F$. If F is rigid but $F-e$ is not rigid for any $e \in E_F$, then F is brace-minimally rigid. However, in this way, we need to carry out rank calculations $O(|E_F|)$ times, whereas our result in Theorem 3 provides a much faster way with just one rank calculation.

3 Proof of Theorem 3

We prove the if part and the only if part of Theorem 3 in Sections 3.1 and 3.2, respectively.

3.1 Proof of the if part

Assume that conditions (a), (b) and (c) are satisfied, i.e., the rank of H_F is $2p$ and G_F is a spanning forest with $2p+1$ connected components. We immediately see that F is rigid by Theorem 2. Consider removing a brace from F. Let F' be the resulting framework. Then $G_{F'}$ is a spanning forest with $2p+2$ connected components. On the other hand, the rank of $H_{F'}$ is at most $2p$ since the number of rows in $H_{F'}$ is $2p$. Therefore by Theorem 2, F' is no longer rigid, which means that F is brace-minimally rigid. This completes the proof of the if part of Theorem 3.

3.2 Proof of the only if part

We show the contrapositive of the only if part: "F is not brace-minimally rigid if one of conditions (a), (b), and (c) is not satisfied."

Case 1: Condition (a) is not satisfied, i.e., the rank of H_F is not $q-1$. Note that the rank of H_F is at most $q-1$ since the sum of all q columns in H_F is zero. Thus, in this case the rank of H_F is less than $q-1$, which means that F is not rigid by Theorem 2.
Case 2: Condition (b) is not satisfied, i.e., G_F has a cycle. In this case we remove a brace corresponding to an edge on the cycle, and let F' be the resulting framework. Since $H_{F'} = H_F$, if F' is rigid, F is not brace-minimally rigid; otherwise, F is not rigid.
Case 3: Condition (c) is not satisfied, i.e., $q \neq 2p+1$. In this case we can assume that both of conditions (a) and (b) are satisfied. We have two subcases [Case 3A] $q > 2p+1$ and [Case 3B] $q < 2p+1$. First consider Case 3A. Recall that the rank of H_F is at most $2p$, which is less than $q-1$ in this subcase. This means that F is not rigid by Theorem 2. In Section 3.2.1, we consider the remaining subcase.

3.2.1 Case 3B

In this section, we consider the case that the rank of H_F is $q-1$ and G_F is a spanning forest with q connected components, where $q < 2p+1$. For this case, we show the existence of a redundant brace in F, i.e., there exists a brace such that the framework is still rigid even if we remove the brace from F.

Let R_F denote the set of rows in H_F. Suppose that $R_F = \{\boldsymbol{r}_1^{\mathrm{h}}, \boldsymbol{r}_1^{\mathrm{v}}, \boldsymbol{r}_2^{\mathrm{h}}, \ldots, \boldsymbol{r}_p^{\mathrm{v}}\}$, where

$$\boldsymbol{r}_k^{\mathrm{h}} = \begin{bmatrix} \alpha_{k1} & \alpha_{k2} & \cdots & \alpha_{kq} \end{bmatrix} \quad \forall k \in \{1,\ldots,p\}, \quad \text{and} \tag{6}$$

$$\boldsymbol{r}_k^{\mathrm{v}} = \begin{bmatrix} \beta_{k1} & \beta_{k2} & \cdots & \beta_{kq} \end{bmatrix} \quad \forall k \in \{1,\ldots,p\}. \tag{7}$$

We first determine a maximal independent set $B \subseteq R_F$ and a row $\boldsymbol{r}^* \in R_F \setminus B$ using the following procedure. Note that since $|B| = q-1 < 2p = |R_F|$, $R_F \setminus B \neq \emptyset$.

1. Choose a maximal independent set $B \subseteq R_F$ and a row $\boldsymbol{r}^* \in R_F \setminus B$ arbitrarily. Suppose $\boldsymbol{r}^* = \boldsymbol{r}_{k^*}^{\mathrm{h}}$ with an integer $k^* \in \{1,\ldots,p\}$.

2. If there exists a left h-strip for hole k^* with at least one brace, the chosen B and $\boldsymbol{r}^*$ are valid; otherwise go to 3.

3. In this case, the leftmost v-bar in every left h-strip for hole k^* is on the boundary of another hole by Lemma 5 (as shown in Figure 4). Call such a hole a *left hole* for hole k^*. In the left holes for hole k^*, if there exists a hole k' such that $\boldsymbol{r}_{k'}^{\mathrm{h}} \in R_F \setminus B$, set $k^* \leftarrow k'$ and $\boldsymbol{r}^* \leftarrow \boldsymbol{r}_{k'}^{\mathrm{h}}$, and go to 2; otherwise go to 4.

4. In this case, any left hole for hole k^*, say hole k', satisfies $\boldsymbol{r}_{k'}^{\mathrm{h}} \in B$. Since $B \setminus \{\boldsymbol{r}_{k'}^{\mathrm{h}}\} \cup \{\boldsymbol{r}_{k*}^{\mathrm{h}}\}$ is also independent by Lemma 6, set $B \leftarrow B \setminus \{\boldsymbol{r}_{k'}^{\mathrm{h}}\} \cup \{\boldsymbol{r}_{k*}^{\mathrm{h}}\}$, $k^* \leftarrow k'$ and $\boldsymbol{r}^* \leftarrow \boldsymbol{r}_{k'}^{\mathrm{h}}$, and go to 2.

Using the above procedure, we obtain $B \subseteq R_F$ and a row $\boldsymbol{r}^* = \boldsymbol{r}_{k^*}^{\mathrm{h}} \in R_F \setminus B$ such that there exists a left h-strip for hole k^* with at least one brace. If $\boldsymbol{r}^* = \boldsymbol{r}_{k^*}^{\mathrm{v}}$ with an integer $k^* \in \{1,\ldots,p\}$ at the first step, we apply the similar procedure, changing "left", "h-" and "v-" to "upper", "v-" and "h-", respectively.

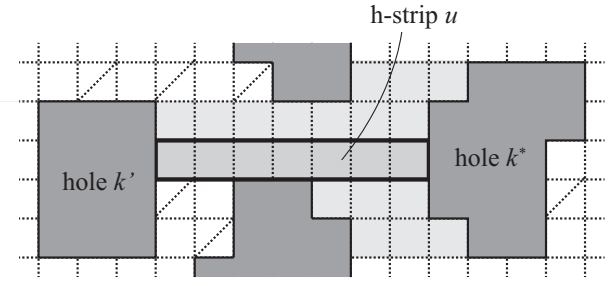


Figure 4: An h-strip u with no brace which is a left h-strip for hole k^* and a right h-strip for hole k' (the light gray part represents all the left h-strips for hole k^* where there is no brace)

Lemma 4 *For a maximal independent set $B \subseteq R_F$ and a row $\boldsymbol{r}^* = \boldsymbol{r}^{\mathrm{h}}_{k^*}$(resp. $\boldsymbol{r}^{\mathrm{v}}_{k^*}$) $\in R_F \setminus B$ with $k^* \in \{1, \ldots, p\}$ which are determined by the above procedure, there exists a left h-strip* (resp. *an upper v-strip*) *for hole k^* with at least one brace.*

We prove Lemmas 5 and 6 which the third and forth steps of procedure are based on, respectively.

Lemma 5 *Given a maximal independent set $B \subseteq R_F$ and a row $\boldsymbol{r}^{\mathrm{h}}_{k^*} \in R_F \setminus B$, suppose that there exists an h-strip $u \in U_F$ with no brace which is a left h-strip for hole k^* (see Figure 4). Then, the leftmost v-bar in h-strip u is on the boundary of another hole.*

Proof: Suppose that the leftmost v-bar in h-strip u is on the boundary of the outer face. In q connected components of G_F, there exists G^l_F with $l \in \{1, \ldots, q\}$ which consists of only u. Looking at the l-th column, $\alpha_{k^* l} = 1$ and all the other entries are zero. This implies that $B \cup \{\boldsymbol{r}^{\mathrm{h}}_{k^*}\}$ is independent, which contradicts the maximality of B. □

Lemma 6 *Given a maximal independent set $B \subseteq R_F$ and a row $\boldsymbol{r}^{\mathrm{h}}_{k^*} \in R_F \setminus B$, suppose that there exists an h-strip $u \in U_F$ with no brace which is a left h-strip for hole k^* and a right h-strip for hole k' (see Figure 4). Then, $B \setminus \{\boldsymbol{r}^{\mathrm{h}}_{k'}\} \cup \{\boldsymbol{r}^{\mathrm{h}}_{k*}\}$ is independent.*

Proof: In q connected components of G_F, there exists G^l_F with $l \in \{1, \ldots, q\}$ which consists of only u. Then, looking at the l-th column, $\alpha_{k^* l} = 1$, $\alpha_{k' l} = -1$ and all the other entries are zero. This implies that $B \setminus \{\boldsymbol{r}^{\mathrm{h}}_{k'}\} \cup \{\boldsymbol{r}^{\mathrm{h}}_{k*}\}$ is independent. □

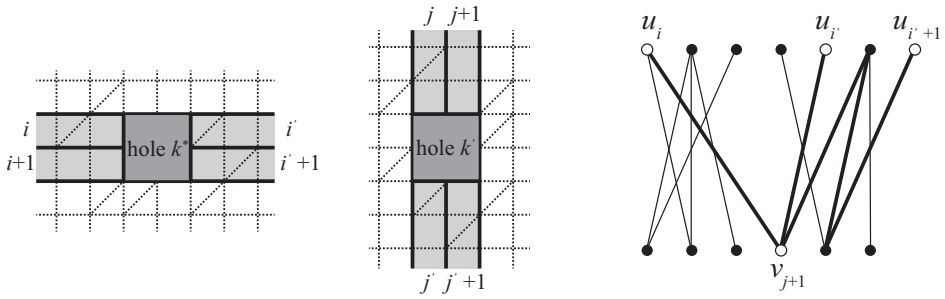
Suppose that $B \subseteq R_F$ and $\boldsymbol{r}^* \in R_F \setminus B$ have been determined by the above procedure. Also consider the unique minimal dependent set $C \subseteq B \cup \{\boldsymbol{r}^*\}$. We then observe that $\boldsymbol{r}^* \in C$ and if $\boldsymbol{r}^* = \boldsymbol{r}^{\mathrm{h}}_{k^*}$ (resp. $\boldsymbol{r}^{\mathrm{v}}_{k^*}$) with $k^* \in \{1, \ldots, p\}$, there exists a left h-strip (resp. upper v-strip) for hole k^* with at least one brace by Lemma 4. Let $G^{l^*}_F$ be the connected component of G_F which such the braced left h-strip (resp. upper v-strip) belongs to. Recall that G_F forms a spanning forest in Case 3B, thus $G^{l^*}_F$ forms a tree. Let $G^{l^*}_F(C)$ denote the minimal subtree of $G^{l^*}_F$ including all vertices in $\bigcup_{\boldsymbol{r} \in C}\{U^{l^*}_F(\boldsymbol{r}) \cup V^{l^*}_F(\boldsymbol{r})\}$, where

$$U^{l^*}_F(\boldsymbol{r}) = \begin{cases} \{u \in U^{l^*}_F \mid u \text{ is a left or right h-strip for hole } k\} & \text{if } \boldsymbol{r} = \boldsymbol{r}^{\mathrm{h}}_k \text{ with } k \in \{1, \ldots, p\} \\ \emptyset & \text{if } \boldsymbol{r} = \boldsymbol{r}^{\mathrm{v}}_k \text{ with } k \in \{1, \ldots, p\}, \end{cases} \tag{8}$$

$$V^{l^*}_F(\boldsymbol{r}) = \begin{cases} \emptyset & \text{if } \boldsymbol{r} = \boldsymbol{r}^{\mathrm{h}}_k \text{ with } k \in \{1, \ldots, p\} \\ \{v \in V^{l^*}_F \mid v \text{ is an upper or lower v-strip for hole } k\} & \text{if } \boldsymbol{r} = \boldsymbol{r}^{\mathrm{v}}_k \text{ with } k \in \{1, \ldots, p\}. \end{cases} \tag{9}$$

We notice that $G^{l^*}_F$ includes at least one brace (i.e., $E^{l^*}_F \neq \emptyset$) and $G^{l^*}_F(C)$ includes at least one vertex.

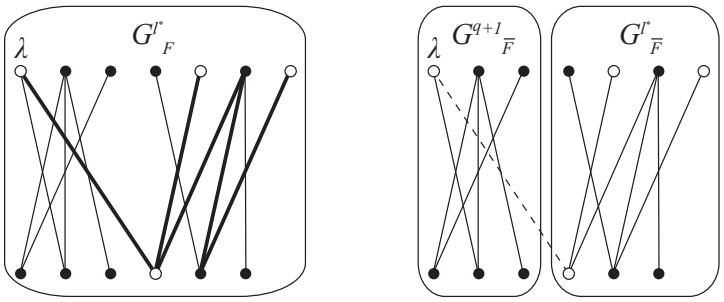
Let us consider an example shown in Figure 5. Suppose $C = \{\boldsymbol{r}^{\mathrm{h}}_{k^*}(= \boldsymbol{r}^*), \boldsymbol{r}^{\mathrm{v}}_{k'}\}$. We then focus on the left and right h-strips for hole k^* and the upper and lower v-strips for hole k'. As shown in Figure 5(a),



(a) the left and right h-strips for hole k^* and the upper and lower v-strips for hole k'

(b) $G^{l^*}_F$

Figure 5: An example with $C = \{\boldsymbol{r}^{\mathrm{h}}_{k^*}(= \boldsymbol{r}^*), \boldsymbol{r}^{\mathrm{v}}_{k'}\}$



(a) $\lambda(= u_i$ in Figure 5(b)) (b) $G^{l^*}_{\bar{F}}$ and $G^{q+1}_{\bar{F}}$ obtained by removing the dashed edge

Figure 6: Illustrations of $G^{l^*}_F$, $G^{l^*}_{\bar{F}}$ and $G^{q+1}_{\bar{F}}$ for the same example in Figure 5

the i-th and $(i+1)$-th (resp. i'-th and $(i'+1)$-th) h-strips lie on the left (resp. right) side of hole k^* and the j-th and $(j+1)$-th (resp. j'-th and $(j'+1)$-th) v-strips lie on the upper (resp. lower) side of hole k' in this example. Here a left h-strip for hole k^*, i.e., the i-th h-strip, is given a brace, we thus determine $G^{l^*}_F$ as a component of G_F which includes vertex u_i (see Figure 5(b)). Suppose that $G^{l^*}_F$ also includes $u_{i'}$, $u_{i'+1}$ and v_{j+1} and does not include u_{i+1}, v_j, $v_{j'}$ and $v_{j'+1}$, i.e., $\bigcup_{\boldsymbol{r}\in C}\{U^{l^*}_F(\boldsymbol{r}) \cup V^{l^*}_F(\boldsymbol{r})\} = \{u_i, u_{i'}, u_{i'+1}, v_{j+1}\}$. In Figure 5(b), white circles represent vertices in $\bigcup_{\boldsymbol{r}\in C}\{U^{l^*}_F(\boldsymbol{r}) \cup V^{l^*}_F(\boldsymbol{r})\}$ and heavy lines represent $G^{l^*}_F(C)$.

In what follows, we show the existence of a redundant brace in $G^{l^*}_F$. Note that every leaf of $G^{l^*}_F(C)$ must be a vertex in $\bigcup_{\boldsymbol{r}\in C}\{U^{l^*}_F(\boldsymbol{r}) \cup V^{l^*}_F(\boldsymbol{r})\}$ (see Figure 6(a)). Choose one of those leaves of $G^{l^*}_F(C)$, say λ. If $G^{l^*}_F(C)$ includes two or more vertices, remove a brace in $G^{l^*}_F(C)$ incident to λ (see Figure 6(b)); otherwise remove an arbitrary brace in $G^{l^*}_F$. Let $\bar{F}$ be the resulting whole framework. Then, for $l \in \{1, \ldots, q\} \setminus \{l^*\}$, the l-th connected component remains in $G_{\bar{F}}$ without any change, i.e., $G^l_{\bar{F}} = G^l_F$. Only $G^{l^*}_F$ is separated into two components $G^{l^*}_{\bar{F}}$ and $G^{q+1}_{\bar{F}}$ so that $G^{q+1}_{\bar{F}}$ includes λ and $G^{l^*}_{\bar{F}}$ does not.

Next let us see hole matrix $H_{\bar{F}}$ consisting of rows $R_{\bar{F}}$. Note that $|R_{\bar{F}}| = |R_F| = 2p$ and each row in $R_{\bar{F}}$ consists of $q+1$ entries. For integers $k \in \{1, \ldots, p\}$ and $l \in \{1, \ldots, q+1\}$, let $\bar{\alpha}^+_{kl}$ (resp. $\bar{\alpha}^-_{kl}$, $\bar{\beta}^+_{kl}$ and $\bar{\beta}^-_{kl}$) be the number of left h-strips (resp. right h-strips, upper v-strips and lower h-strips) for hole k belonging to $G^l_{\bar{F}}$. Let $\bar{\alpha}_{kl} = \bar{\alpha}^+_{kl} - \bar{\alpha}^-_{kl}$ and $\bar{\beta}_{kl} = \bar{\beta}^+_{kl} - \bar{\beta}^-_{kl}$, respectively. For the h-th row $\boldsymbol{r} \in R_F$ with $h \in \{1, \ldots, 2p\}$, let $\rho(\boldsymbol{r})$ denote the h-th row in $R_{\bar{F}}$. Then $R_{\bar{F}} = \{\rho(\boldsymbol{r}^{\rm h}_1), \rho(\boldsymbol{r}^{\rm v}_1), \rho(\boldsymbol{r}^{\rm h}_2), \ldots, \rho(\boldsymbol{r}^{\rm v}_p)\}$, where

$$\rho(\boldsymbol{r}^{\rm h}_k) = \begin{bmatrix} \bar{\alpha}_{k1} & \bar{\alpha}_{k2} & \cdots & \bar{\alpha}_{kq} & \bar{\alpha}_{k,q+1} \end{bmatrix} \quad \forall k \in \{1, \ldots, p\}, \quad \text{and} \tag{10}$$

$$\rho(\boldsymbol{r}^{\rm v}_k) = \begin{bmatrix} \bar{\beta}_{k1} & \bar{\beta}_{k2} & \cdots & \bar{\beta}_{kq} & \bar{\beta}_{k,q+1} \end{bmatrix} \quad \forall k \in \{1, \ldots, p\}. \tag{11}$$

In the following, for $R' \subseteq R_F$, we abuse the notation $\rho(R')$ to denote the subset of $R_{\bar{F}}$, $\{\rho(\boldsymbol{r}) \mid \boldsymbol{r} \in R'\}$. In addition, we use the notation $\varepsilon_l(\boldsymbol{r})$ to denote the l-th entry of row $\boldsymbol{r}$, e.g., $\varepsilon_l(\boldsymbol{r}^{\rm h}_k) = \alpha_{kl}$ and $\varepsilon_l(\rho(\boldsymbol{r}^{\rm v}_k)) = \bar{\beta}_{kl}$. Let us see the following remark.

Remark 7 *For any $\boldsymbol{r} \in R_F$ and $l \in \{1, \ldots, q\} \setminus \{l^*\}$, $\varepsilon_l(\rho(\boldsymbol{r})) = \varepsilon_l(\boldsymbol{r})$.*

Recall that λ is the leaf of $G^{l^*}_F(C)$ which is included in $G^{q+1}_{\bar{F}}$. Let $\tilde{\boldsymbol{r}}$ be a row in C such that $\lambda \in U^{l^*}_F(\tilde{\boldsymbol{r}}) \cup V^{l^*}_F(\tilde{\boldsymbol{r}})$. We then show the following lemma.

Lemma 8 *(i) For $\boldsymbol{r} \in R_F$, $\varepsilon_{l^*}(\rho(\boldsymbol{r})) + \varepsilon_{q+1}(\rho(\boldsymbol{r})) = \varepsilon_{l^*}(\boldsymbol{r})$. (ii) $\varepsilon_{l^*}(\rho(\tilde{\boldsymbol{r}})) = \varepsilon_{l^*}(\tilde{\boldsymbol{r}}) - \delta$ and $\varepsilon_{q+1}(\rho(\tilde{\boldsymbol{r}})) = \delta$, where $\delta = -1$ or 1. (iii) For $\boldsymbol{r} \in C \setminus \{\tilde{\boldsymbol{r}}\}$, $\varepsilon_{l^*}(\rho(\boldsymbol{r})) = \varepsilon_{l^*}(\boldsymbol{r})$ and $\varepsilon_{q+1}(\rho(\boldsymbol{r})) = 0$.*

Proof: (i) immediately follows the fact that $G^{l^*}_F$ is separated into $G^{l^*}_{\bar{F}}$ and $G^{q+1}_{\bar{F}}$. Suppose that $\tilde{\boldsymbol{r}} = \boldsymbol{r}^{\rm h}_{\tilde{k}}$ with $\tilde{k} \in \{1, \ldots, p\}$. Then λ is a left or right h-strip for hole $\tilde{k}$. If λ is a left h-strip for hole $\tilde{k}$, we have $\bar{\alpha}^+_{\tilde{k}l^*} = \alpha^+_{\tilde{k}l^*} - 1$, $\bar{\alpha}^-_{\tilde{k}l^*} = \alpha^-_{\tilde{k}l^*}$, $\bar{\alpha}^+_{\tilde{k},q+1} = 1$, and $\bar{\alpha}^-_{\tilde{k},q+1} = 0$, i.e., $\bar{\alpha}_{\tilde{k}l^*} = \alpha_{\tilde{k}l^*} - 1$ and $\bar{\alpha}_{\tilde{k},q+1} = 1$; otherwise $\bar{\alpha}_{\tilde{k}l^*} = \alpha_{\tilde{k}l^*} + 1$ and $\bar{\alpha}_{\tilde{k},q+1} = -1$. Similarly, for the case that $\tilde{\boldsymbol{r}} = \boldsymbol{r}^{\rm v}_{\tilde{k}}$ with $\tilde{k} \in \{1, \ldots, p\}$, we can prove

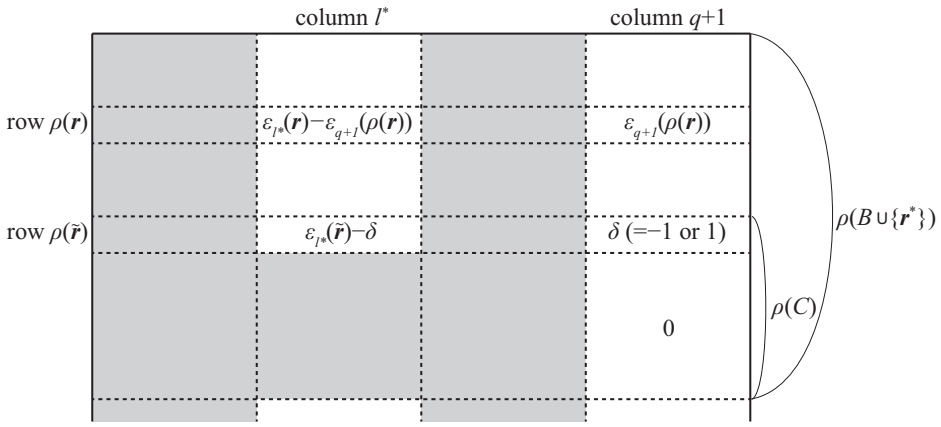


Figure 7: $H_{\bar{F}}$ (each entry in the dark part is the same as in H_F)

$\bar{\beta}_{\tilde{k}l^*} = \beta_{\tilde{k}l^*} \mp 1$ and $\bar{\beta}_{\tilde{k},q+1} = \pm 1$, thus (ii) is proved. For $\boldsymbol{r}_k^{\mathrm{h}}$ (resp. $\boldsymbol{r}_k^{\mathrm{v}}$) $\in C \setminus \{\tilde{\boldsymbol{r}}\}$, any left or right h-strip (resp. upper or lower v-strip) for hole k belonging to $G_F^{l^*}$ remains in $G_{\bar{F}}^{l^*}$, thus (iii) holds. □

We represent the statements of Remark 7 and Lemma 8 as an illustration in Figure 7.

In the rest of this section, we show that $\bar{F}$ is rigid, i.e., F is not brace-minimally rigid. Let us first confirm the following remark.

Remark 9 *$B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}$ is independent.*

We then show the following lemma.

Lemma 10 *$\rho(B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\})$ is independent.*

Proof: We prove by contradiction. Suppose that $\rho(B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\})$ is dependent:

$$\sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \mu_{\boldsymbol{r}} \cdot \rho(\boldsymbol{r}) = \boldsymbol{0}, \tag{12}$$

where $\mu_{\boldsymbol{r}}$ is a real number such that $\mu_{\boldsymbol{r}} \neq 0$ $\exists \boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}$. The left-hand side of (12) can be represented as

$$\sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \mu_{\boldsymbol{r}} \cdot \left[\begin{array}{ccccccc} \varepsilon_1(\rho(\boldsymbol{r})) & \cdots & \varepsilon_{l^*}(\rho(\boldsymbol{r})) & \cdots & \varepsilon_q(\rho(\boldsymbol{r})) & \varepsilon_{q+1}(\rho(\boldsymbol{r})) \end{array} \right]. \tag{13}$$

By Remark 7 and Lemma 8(ii), (13) is equal to

$$\begin{aligned} &\sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \mu_{\boldsymbol{r}} \cdot \left[\begin{array}{ccccccc} \varepsilon_1(\boldsymbol{r}) & \cdots & \varepsilon_{l^*}(\boldsymbol{r}) & \cdots & \varepsilon_q(\boldsymbol{r}) & 0 \end{array} \right] \\ &\qquad + \sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \mu_{\boldsymbol{r}} \cdot \left[\begin{array}{ccccccc} 0 & \cdots & -\varepsilon_{q+1}(\rho(\boldsymbol{r})) & \cdots & 0 & \varepsilon_{q+1}(\rho(\boldsymbol{r})) \end{array} \right]. \end{aligned} \tag{14}$$

Looking at the $(q+1)$-th entries in (14), the sum of those entries is zero by (12), i.e.,

$$\sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \mu_{\boldsymbol{r}} \cdot \varepsilon_{q+1}(\rho(\boldsymbol{r})) = 0. \tag{15}$$

This means that the second summation term in (14) is a zero vector, and therefore the first term is also a zero vector. We thus have

$$\sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \mu_{\boldsymbol{r}} \cdot \boldsymbol{r} = \boldsymbol{0}, \tag{16}$$

which contradicts the independency of $B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}$ shown in Remark 9. □

The following lemma implies the rigidity of $\bar{F}$.

Lemma 11 *$\rho(B \cup \{\boldsymbol{r}^*\})$ is independent.*

Proof: We prove by contradiction. Suppose that $\rho(B \cup \{\boldsymbol{r}^*\})$ is dependent. Since $\rho(B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\})$ is independent by Lemma 10, any dependent subset of $\rho(B \cup \{\boldsymbol{r}^*\})$ includes $\rho(\tilde{\boldsymbol{r}})$. Thus, $\rho(\tilde{\boldsymbol{r}})$ is represented as a linear combination of rows in $\rho(B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\})$:

$$\rho(\tilde{\boldsymbol{r}}) = \sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \phi_{\boldsymbol{r}} \cdot \rho(\boldsymbol{r}), \tag{17}$$

where $\phi_{\boldsymbol{r}}$ is a real number such that $\phi_{\boldsymbol{r}} \neq 0$ $\exists \boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}$. Equation (17) can be deformed as

$$\begin{aligned} &\left[\begin{array}{ccccccc} \varepsilon_1(\rho(\tilde{\boldsymbol{r}})) & \cdots & \varepsilon_{l^*}(\rho(\tilde{\boldsymbol{r}})) & \cdots & \varepsilon_q(\rho(\tilde{\boldsymbol{r}})) & \varepsilon_{q+1}(\rho(\tilde{\boldsymbol{r}})) \end{array} \right] \\ &\qquad = \sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \phi_{\boldsymbol{r}} \cdot \left[\begin{array}{ccccccc} \varepsilon_1(\rho(\boldsymbol{r})) & \cdots & \varepsilon_{l^*}(\rho(\boldsymbol{r})) & \cdots & \varepsilon_q(\rho(\boldsymbol{r})) & \varepsilon_{q+1}(\rho(\boldsymbol{r})) \end{array} \right]. \end{aligned} \tag{18}$$

By Remark 7 and Lemma 8(ii), the left-hand side of (18) can be represented as

$$\left[\begin{array}{ccccccc} \varepsilon_1(\tilde{\boldsymbol{r}}) & \cdots & \varepsilon_{l^*}(\tilde{\boldsymbol{r}}) & \cdots & \varepsilon_q(\tilde{\boldsymbol{r}}) & 0 \end{array} \right] + \left[\begin{array}{ccccccc} 0 & \cdots & -\delta & \cdots & 0 & \delta \end{array} \right], \tag{19}$$

where $\delta = -1$ or 1. Similarly, by Remark 7 and Lemma 8(i), the right-hand side of (18) can be represented as

$$\begin{aligned} &\sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \phi_{\boldsymbol{r}} \cdot \left[\begin{array}{ccccccc} \varepsilon_1(\boldsymbol{r}) & \cdots & \varepsilon_{l^*}(\boldsymbol{r}) & \cdots & \varepsilon_q(\boldsymbol{r}) & 0 \end{array} \right] \\ &\qquad + \sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \phi_{\boldsymbol{r}} \cdot \left[\begin{array}{ccccccc} 0 & \cdots & -\varepsilon_{q+1}(\rho(\boldsymbol{r})) & \cdots & 0 & \varepsilon_{q+1}(\rho(\boldsymbol{r})) \end{array} \right]. \end{aligned} \tag{20}$$

Looking at the $(q+1)$-th entries in (18), (19) and (20), we obtain

$$\delta = \sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \phi_{\boldsymbol{r}} \cdot \varepsilon_{q+1}(\rho(\boldsymbol{r})), \tag{21}$$

which means that the second terms in (19) and (20) are equivalent, and therefore the first terms are also equivalent. We thus have

$$\tilde{\boldsymbol{r}} = \sum_{\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}} \phi_{\boldsymbol{r}} \cdot \boldsymbol{r}. \tag{22}$$

Recall that $B \cup \{\boldsymbol{r}^*\}$ includes the unique minimal dependent set of rows, which is C, and $\tilde{\boldsymbol{r}} \in C$. Because of the uniqueness and the minimality of C, $\phi_{\boldsymbol{r}}$ is uniquely determined for $\boldsymbol{r} \in B \cup \{\boldsymbol{r}^*\} \setminus \{\tilde{\boldsymbol{r}}\}$ such that

$$\phi_{\boldsymbol{r}} \neq 0 \qquad \forall \boldsymbol{r} \in C \setminus \{\tilde{\boldsymbol{r}}\}, \quad \text{and} \tag{23}$$

$$\phi_{\boldsymbol{r}} = 0 \qquad \forall \boldsymbol{r} \in B \setminus C. \tag{24}$$

By (23) and (24), equation (21) can be deformed as

$$\delta = \sum_{\boldsymbol{r} \in C \setminus \{\tilde{\boldsymbol{r}}\}} \phi_{\boldsymbol{r}} \cdot \varepsilon_{q+1}(\rho(\boldsymbol{r})). \tag{25}$$

Here $\varepsilon_{q+1}(\rho(\boldsymbol{r})) = 0$ for any $\boldsymbol{r} \in C \setminus \{\tilde{\boldsymbol{r}}\}$ by Lemma 8(iii), thus the right-hand side of (25) is equal to zero, whereas $\delta = -1$ or 1, contradiction. This concludes the proof. □

We now observe that the rank of $H_{\bar{F}}$ is at least q since $\rho(B \cup \{\boldsymbol{r}^*\})$ consists of q independent rows by Lemma 11. Note that the rank of $H_{\bar{F}}$ is at most q since the sum of all $q+1$ columns in $H_{\bar{F}}$ is zero, so the rank of $H_{\bar{F}}$ is q. This means that $\bar{F}$ is rigid by Theorem 2, i.e., F is not brace-minimally rigid. We thus complete the proof of Theorem 3.

References

[1] E. D. Bolker and H. Crapo, "Bracing rectangular frameworks", *SIAM Journal on Applied Mathematics*, 36(3), pp. 473–490 (1979).

[2] Z. Gáspár, N. Radics and A. Recski, "Rigidity of square grids with holes", *Computer Assisted Mechanics and Engineering Sciences*, 6(3–4), pp. 329–335 (1999).

[3] Y. Ito, Y. Kobayashi, Y. Higashikawa, N. Katoh, S. H. Poon, and M. Saumell, "Optimally bracing grid frameworks with holes", *Theoretical Computer Science*, 607, pp. 337–350 (2015).

[4] G. Laman, "On graphs and rigidity of plane skeletal structures", *Journal of Engineering Mathematics*, 4(4), pp. 331–340 (1970).

[5] J. C. Maxwell, "On the calculation of the equilibrium and stiffness of frames", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 27(182), pp. 294–299 (1864).

[6] N. Radics and A. Recski, "Applications of combinatorics to statics—rigidity of grids", *Discrete Applied Mathematics*, 123(1), pp. 473–485 (2002).

[7] W. Whiteley, "Some matroids from discrete applied geometry", *Contemporary Mathematics*, 197, pp. 171–312 (1996).

An impossibility theorem for paired comparisons

László Csató[1]

Laboratory on Engineering and Management
Intelligence, Research Group of Operations
Research and Decision Systems
Institute for Computer Science and Control,
Hungarian Academy of Sciences (MTA SZTAKI)
Department of Operations Research and
Actuarial Sciences
Corvinus University of Budapest (BCE)
1093 Budapest, Fővám tér 8., Hungary
laszlo.csato@uni-corvinus.hu

Abstract: In several decision-making problems, alternatives should be ranked on the basis of paired comparisons between them. An axiomatic approach for the universal ranking problem with arbitrary preference intensities, incomplete and multiple comparisons is presented. In particular, two basic properties – independence of irrelevant matches and self-consistency – are considered. It is revealed that there exists no ranking method satisfying both requirements at the same time. The impossibility result holds under various restrictions on the set of ranking problems, however, it does not emerge in the case of round-robin tournaments. An interesting and more general possibility result is obtained by weakening independence of irrelevant matches through the concept of macrovertex.

Keywords: Preference aggregation; paired comparison; tournament ranking; axiomatic approach; impossibility

1 Introduction

Let $N = \{X_1, X_2, \ldots, X_n\}$, $n \in \mathbb{N}$ be the *set of objects* and $T = [t_{ij}] \in \mathbb{R}^{n \times n}$ be a *tournament matrix* such that $t_{ij} + t_{ji} \in \mathbb{N}$. t_{ij} represents the aggregated score of object X_i against X_j, $t_{ij}/(t_{ij} + t_{ji})$ can be interpreted as the likelihood that object X_i is better than object X_j. $t_{ii} = 0$ is assumed for all $X_i \in N$.

The pair (N, T) is called a *ranking problem*. The set of ranking problems with $|N| = n$ is denoted by $\mathcal{R}^n$.

A *scoring procedure* f is an $\mathcal{R}^n \to \mathbb{R}^n$ function giving a rating for each object. Any scoring method immediately induces a ranking (a transitive and complete weak order on the set $N \times N$) $\succeq$ by $f_i(N, T) \geq f_j(N, T)$ meaning that X_i is ranked weakly above X_j, denoted by $X_i \succeq X_j$. Every scoring method can be considered as a *ranking method*. This paper discusses only axioms for ranking methods induced by scoring procedures.

A ranking problem (N, T) is associated with the skew-symmetric *results matrix* $R = T - T^\top = [r_{ij}] \in \mathbb{R}^{n \times n}$ and the symmetric *matches matrix* $M = T + T^\top = [m_{ij}] \in \mathbb{N}^{n \times n}$ such that m_{ij} is the number of the comparisons between X_i and X_j, whose outcome is given by r_{ij}. Matrices R and M also determine the tournament matrix as $T = (R + M)/2$.

Remark 1 *Any ranking problem $(N, T) \in \mathcal{R}^n$ can be denoted analogously by (N, R, M) with the restriction $|r_{ij}| \leq m_{ij}$ for all $X_i, X_j \in N$, that is, the outcome of any paired comparison between two objects cannot 'exceed' their number of matches.*

[1] The research is supported by OTKA grant K 111797 and by the MTA Premium Post Doctorate Research Program.

Definition 2 *A ranking problem $(N, R, M) \in \mathcal{R}^n$ is called*

- balanced *if* $\sum_{X_k \in N} m_{ik} = \sum_{X_k \in N} m_{jk}$ *for all* $X_i, X_j \in N$*;*
- round-robin *if* $m_{ij} = m_{k\ell}$ *for all* $X_i \neq X_j$ *and* $X_k \neq X_\ell$*;*
- unweighted *if* $m_{ij} \in \{0; 1\}$ *for all* $X_i, X_j \in N$*;*
- extremal *if* $|r_{ij}| \in \{0; m_{ij}\}$ *for all* $X_i, X_j \in N$*.*

The set of balanced ranking problems is denoted by $\mathcal{R}_B$.
The set of round-robin ranking problems is denoted by $\mathcal{R}_R$.
The set of unweighted ranking problems is denoted by $\mathcal{R}_U$.
The set of extremal ranking problems is denoted by $\mathcal{R}_E$.

The *maximal number of comparisons* is $m = \max_{X_i, X_j \in N} m_{ij}$.

Axiom 3 Independence of irrelevant matches *(IIM): Let $(N, T), (N, T') \in \mathcal{R}^n$ be two ranking problems and $X_i, X_j, X_k, X_\ell \in N$ be four different objects such that (N, T) and (N, T') are identical but $t'_{k\ell} \neq t_{k\ell}$. Scoring procedure $f : \mathcal{R}^n \to \mathbb{R}^n$ is called* independent of irrelevant matches *if $f_i(N, T) \geq f_j(N, T) \Rightarrow f_i(N, T') \geq f_j(N, T')$.*

Definition 4 Opponent set*: Let $(N, R, M) \in \mathcal{R}^n_U$ be an unweighted ranking problem. The* opponent set *of object X_i is $O_i = \{X_j : m_{ij} = 1\}$*

Objects of the opponent set O_i are called *opponents* of X_i.

Definition 5 *Let $(N, R, M) \in \mathcal{R}^n_U$ be an unweighted ranking problem, $X_i, X_j \in N$ be two different objects and $g : O_i \leftrightarrow O_j$ be a one-to-one correspondence between the opponents of X_i and X_j. Then $\mathfrak{g} : \{k : X_k \in O_i\} \leftrightarrow \{\ell : X_\ell \in O_j\}$ is given by $X_{\mathfrak{g}(k)} = g(X_k)$.*

Definition 6 Sum of ranking problems*: Let $(N, R, M), (N, R', M') \in \mathcal{R}^n$ be two ranking problems with the same object set N. The* sum *of these ranking problems is the ranking problem $(N, R + R', M + M') \in \mathcal{R}^n$.*

Definition 6 implies that any ranking problem can be decomposed into unweighted ranking problems, that is, it can be obtained as a sum of unweighted ranking problems. However, while the sum of ranking problems is unique, a ranking problem may have a number of possible decompositions.

Axiom 7 Self-consistency *(SC) [2]: Let $(N, R, M) \in \mathcal{R}^n$ be a ranking problem such that $R = \sum_{p=1}^m R^{(p)}$, $M = \sum_{p=1}^m M^{(p)}$ and $(N, R^{(p)}, M^{(p)}) \in \mathcal{R}^n_U$ is an unweighted ranking problem for all $p = 1, 2, \ldots, m$. Let $X_i, X_j \in N$ be two objects and $f : \mathcal{R}^n \to \mathbb{R}^n$ be a scoring procedure such that for all $p = 1, 2, \ldots, m$ there exists a one-to-one mapping $g^{(p)}$ from $O_i^{(p)}$ onto $O_j^{(p)}$, where $r_{ik}^{(p)} \geq r_{j\mathfrak{g}^{(p)}(k)}^{(p)}$ and $f_k(N, R, M) \geq f_{\mathfrak{g}^{(p)}(k)}(N, R, M)$.*
f is called self-consistent *if $f_i(N, R, M) \geq f_j(N, R, M)$, furthermore, $f_i(N, R, M) > f_j(N, R, M)$ if at least one of the above inequalities is strict.*

2 Main result

Theorem 8 *There does not exist a scoring procedure that is independent of irrelevant matches and self-consistent on the set of ranking problems with at least four objects $\mathcal{R}^n | n \geq 4$.*

Proof: See [3]. □

Lemma 9 *There exist scoring procedures that are independent of irrelevant matches and self-consistent on the set of ranking problems with at most three objects* $\mathcal{R}^n|n \leq 3$.

Lemma 10 *There does not exist a scoring procedure that is independent of irrelevant matches and self-consistent on the set of balanced, unweighted and extremal ranking problems with four objects* $\mathcal{R}^4_B \cap \mathcal{R}^4_U \cap \mathcal{R}^4_E$.

Proposition 11 *There exist scoring procedures that are independent of irrelevant matches and self-consistent on the set of round-robin ranking problems* $\mathcal{R}_R$.

Proof: See [3]. □

Definition 12 Macrovertex *[1, Definition 3.1]: Let* $(N, R, M) \in \mathcal{R}^n$ *be a ranking problem. Object set* $V \subseteq N$ *is called* macrovertex *if* $m_{ik} = m_{jk}$ *for all* $X_i, X_j \in V$ *and* $X_k \in N \setminus V$.

Axiom 13 Macrovertex independence *(MVI) [1, Property 8]: Let* $V \subseteq N$ *be a macrovertex in ranking problems* $(N, T), (N, T') \in \mathcal{R}^n$ *and* $X_i, X_j \in V$ *be two different objects such that* (N, T) *and* (N, T') *are identical but* $t'_{ij} \neq t_{ij}$. *Scoring procedure* $f : \mathcal{R}^n \to \mathbb{R}^n$ *is called* macrovertex independent *if* $f_k(N, T) \geq f_\ell(N, T) \Rightarrow f_k(N, T') \geq f_\ell(N, T')$ *for all* $X_k, X_\ell \in N \setminus V$.

Axiom 14 Macrovertex autonomy *(MVA): Let* $V \subseteq N$ *be a macrovertex in ranking problems* (N, T), $(N, T') \in \mathcal{R}^n$ *and* $X_k, X_\ell \in N \setminus V$ *be two different objects such that* (N, T) *and* (N, T') *are identical but* $t'_{k\ell} \neq t_{k\ell}$. *Scoring procedure* $f : \mathcal{R}^n \to \mathbb{R}^n$ *is called* macrovertex autonom *if* $f_i(N, T) \geq f_j(N, T) \Rightarrow f_i(N, T') \geq f_j(N, T')$ *for all* $X_i, X_j \in V$.

Proposition 15 *There exist scoring procedures that are macrovertex autonom, macrovertex independent and self-consistent.*

Proof: See [3]. □

Let $(N, R, M) \in \mathcal{R}^n_R$ be a round-robin ranking problem. Then object set $V \subseteq N$ is a macrovertex.

Corollary 16 *MVA / MVI implies IIM on the domain of round-robin raking problems* $\mathcal{R}_R$.

Hence Proposition 15 is more general than Proposition 11 due to Corollary 16.

Axiom 17 Weak self-consistency *(WSC): Let* $(N, R, M) \in \mathcal{R}^n$ *be a ranking problem such that* $R = \sum_{p=1}^m R^{(p)}$, $M = \sum_{p=1}^m M^{(p)}$ *and* $(N, R^{(p)}, M^{(p)}) \in \mathcal{R}^n_U$ *is an unweighted ranking problem for all* $p = 1, 2, \ldots, m$. *Let* $X_i, X_j \in N$ *be two objects and* $f : \mathcal{R}^n \to \mathbb{R}^n$ *be a scoring procedure such that for all* $p = 1, 2, \ldots, m$ *there exists a one-to-one mapping* g *from* $O_i^{(p)}$ *onto* $O_j^{(p)}$, *where* $r_{ik}^{(p)} \geq r_{j\mathfrak{g}^{(p)}(k)}^{(p)}$ *and* $f_k(N, R, M) \geq f_{\mathfrak{g}^{(p)}(k)}(N, R, M)$.
f *is called* weakly self-consistent *if* $f_i(N, R, M) \geq f_j(N, R, M)$, *furthermore,* $f_i(N, R, M) > f_j(N, R, M)$ *if* $r_{ik}^{(p)} > r_{j\mathfrak{g}^{(p)}(k)}^{(p)}$ *for at least one* $p = 1, 2, \ldots, m$.

Proposition 18 *There exist a scoring procedure that is independent of irrelevant matches and weakly self-consistent.*

Proof: See [3]. □

For an extensive discussion of axioms and results, see [3].

Selected references

[1] P. Chebotarev, Aggregation of preferences by the generalized row sum method. *Mathematical Social Sciences* **27**(3):293–320 (1994)

[2] P. Chebotarev and E. Shamis, Constructing an objective function for aggregating incomplete preferences. In Tangian, A. and Gruber, J., editors, *Constructing Scalar-Valued Objective Functions*, volume 453 of *Lecture Notes in Economics and Mathematical Systems*, pages 100–124. Springer, Berlin-Heidelberg (1997)

[3] L. Csató, An impossibility theorem for paired comparisons, *arXiv* **1612.00186** (2016)

New algorithms for cake cutting with equal and unequal shares

Ágnes Cseh

Institute of Economics
Hungarian Academy of Sciences and
Corvinus University of Budapest
Budaörsi út 45., Budapest 1112, Hungary
cseh.agnes@krtk.mta.hu

Tamás Fleiner

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
3 Műegyetem rkp., Budapest 1111, Hungary
fleiner@cs.bme.hu

Erzsébet Romsics

MSc Student
Faculty of Natural Sciences
Budapest University of Technology and
Economics
3 Műegyetem rkp., Budapest 1111, Hungary
romsics.csore@gmail.com

Abstract: An unceasing problem of our prevailing society is the fair division of goods. The problem of fair cake cutting is dividing a heterogeneous and divisible resource, the cake, among k players who value pieces according to their own measure function. The goal is to assign each player a not necessarily connected part of the cake that the player evaluates at least a much as her equal fair share.

In this paper, we present two new algorithms. Our first algorithm guarantees a piece strictly larger than $\frac{1}{k}$ of the whole cake for certain measure functions. In the second setting, we investigate the problem of unequal shares, where each player needs to be assigned a piece that she evaluates at least a much as her predefined fair share. We present an algorithm that delivers such a solution and in some cases, runs faster than all known algorithms. In both problems, we establish upper bounds on the number of cuts our algorithms execute.

Keywords: fair division, cake cutting, strong fair division, unequal shares

1 Introduction

In cake cutting problems, the cake symbolizes a heterogeneous and divisible resource that is to be distributed among k players. Each player has an own measure function that is a finitely additive measure defined in the general way. This measure determines the value of any part of the cake, which can differ from player to player. The aim of fair cake division is to give each player a piece that is worth at least as much as her fair share, evaluated with her own measure function. If shares are meant to be *equal* for all players, then the fair share is defined as $\frac{1}{k}$ of the whole cake. This problem is also known as proportional cake-cutting. The *unequal share* version of the problem (or proportional cake-cutting with different entitlements) is where fair share is defined as a specific demand, given for each participant in the input and summing up to 1 in total.

[1]Supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2016-3/2016) and OTKA grant K108383.
[2]Supported by the Hungarian Scientific Research Fund - OTKA no. K108383.
[3]Supported by the Hungarian Scientific Research Fund - OTKA no. K108673.

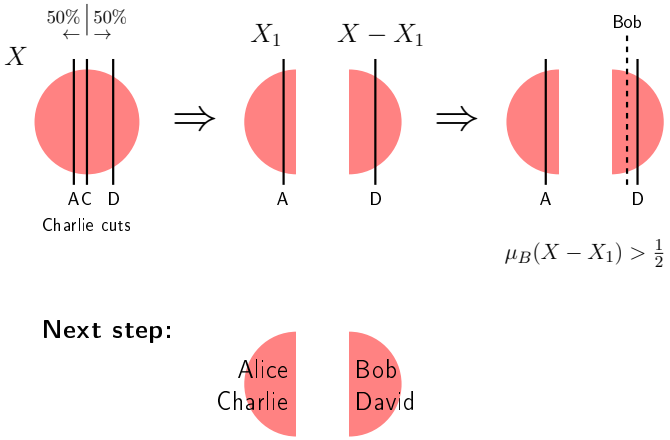


Figure 1: A reduction step in the 4-player Divide and Conquer Algorithm.

1.1 Equal shares

Possibly the most famous fair division method belongs to the class of Divide and Conquer algorithms. The simplest variant called Cut and Choose is a 2-player method that guarantees equal shares. It already appeared in the Old Testament, where Abraham divided the Canaan to two equally valuable parts and his brother Lot chose the one he valued more for himself. A k-player variant of this algorithm was analyzed by Dubins and Spanier [1] and a slightly different version by Even and Paz [3]. The latter show that their method requires $O(k \log k)$ cuts at most, which was later proved to be the best running time that any deterministic algorithm for the equal share cake cutting problem can achieve [2, 7].

We now sketch the k-player variant of the Divide and Conquer Algorithm of Even and Paz and illustrate it on an example with four players. This is a recursive algorithm that reduces the input size to half in each round. We need a so-called passive player, whose role is distinguished, as we will see below. Our example has an even number of players and we remark that an odd number of players can be handled by some minor technical modifications.

Example 1 *Our players are Alice, Bob, Charlie and David, among whom Bob is the sole passive player. Every active player marks the cake with a parallel line where they would divide it into halves, see Figure 1. After that, we choose the mark in the middle (Charlie's in our example) and cut the cake along it. After this, the passive player Bob chooses a side to divide further. He will share the chosen half with the player whose mark is inside it and the remaining two players will restrict themselves to the other half of the cake. With this we have now reduced the 4-player problem to two 2-player problems.*

1.2 Unequal shares

Known methods for the case of unequal shares include cloning players, using Ramsey partitions and most importantly, the so-called Cut Near-Halves Algorithm [4]. The last method computes a fair solution for 2 players with demands d_1 and d_2 in $\lceil \log_2(d_1 + d_2) \rceil$ time. As Robertson and Webb's Algorithm for More Than Two Players (Unequal Shares) [4, page 46] shows, this method can be generalized to k players with demands $d_1, d_2 \ldots d_k$ in the following recursive manner. We assume that $k-1$ players have already divided the cake of value $n = d_1 + d_2 + \cdots + d_k$, the sum of all demands of the players. The last player then challenges each of the first $k-1$ players separately to redistribute the piece already assigned to them. In these rounds, the last player claims $\frac{d_k}{d_1+d_2+\cdots+d_k}$ part of each piece. This generates $k-1$ rounds, each with 2-players. The number of cuts in this algorithm is $O(k^2 \log n)$.

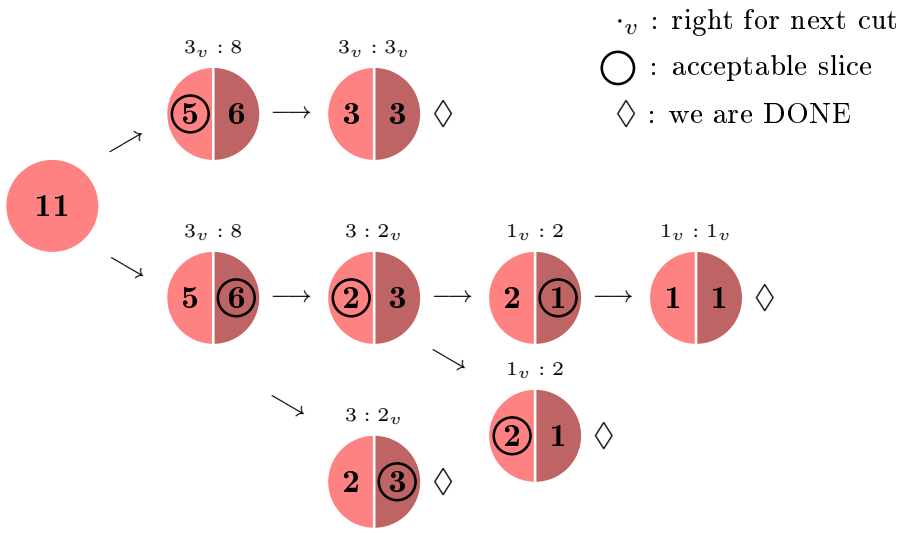


Figure 2: Fair cake cutting in ratio 3 : 8 with the Cut Near-Halves Algorithm

Cut-Near Halves is a simple procedure, in which the cake of an integer value is repeatedly cut in approximately half as follows. The player who claims the lesser portion of the cake cuts the cake into two near-halves, more precisely, if the cake has an odd total value $n = 2\ell + 1$, than she cuts it in ratio $\ell : \ell + 1$, otherwise she cuts it in ratio $1 : 1$. The player who claims the greater portion then picks the piece that she values at least as much as the cutting player. This piece is awarded to this 2nd player and her claim is reduced accordingly. In the next round, the same is repeated on the remaining part of the cake, and so on, until a non-divisible piece of the cake remains. Notice that the player who cuts might change from round to round. The following example illustrates the Cut Near-Halves Algorithm for two players.

Example 2 *Alice and Bob are to divide the cake in ratio 3:8. Alice (the player to get the smaller portion) cuts near-halves in the ratio 5:6. After this, Alice labels each piece with the value she associates to them. Now Bob takes his choice of piece at the value Alice assigned to it. If he takes the piece Alice valued to 6 units, he then has 2 more units to gather, so the other piece must be divided in the reduced ratio 3:2. Now Bob cuts near-halves in the ratio 3:2 and labels the pieces. If Alice chooses the piece valued to 3 units, then we are done, otherwise just continue cutting near-halves. Figure 2 shows the decision tree of this example.*

1.3 Our contribution

After formalizing the most crucial concepts in Section 2 we present two new algorithms in this paper, one for the case of equal shares in Section 3, the other one for the case of unequal shares in Section 4. Due to space restrictions, we have omitted all proofs from this paper.

The motivation to construct our equally dividing Happiness in Unity Algorithm was to develop the principles of the optimal Divide and Conquer Algorithm [4, page 27] further. A property of this previously described algorithm is that the passive player has a chance to get larger piece than her fair share. Our algorithm eliminates the role of the passive player and therefore, it gives every player equal chance to get more than their fair share. A cake division is called *strong fair* if it guarantees for everybody more than her fair share. Our method computes a strong fair division if everyone decides to cut the cake in different places in the first round. This attractive property is achieved by expanding the number of cuts in the

Divide and Conquer Algorithm with a linear term hence it does not change the best reachable $O(k \log k)$ complexity.

Our second result is the k-player Joint-Stock Company Splitting Algorithm on a cake of total value n for the k-player unequal cake division problem. The main difference between our algorithm and the Cut Near-Halves Algorithm [4] is that we swap the roles of the cutter and the chooser. In our algorithm, the player demanding more cuts the cake and the other player chooses. This method is arguable more acceptable form a social point of view. The number of cuts in our algorithm can be bounded from above by $O(2^k \log n)$, which is more than the number of cuts that the Algorithm for More Than Two Players (Unequal Shares) needs. Our future research objective is thus to tighten our estimate for the number of cuts so as to achieve the complexity of the Algorithm for More Than Two Players (Unequal Shares). A preliminary version of our results is contained in [5, 6].

2 Preliminaries

First we formally define our input. Our setting includes a set of players denoted by $\{P_1, P_2, \ldots, P_k\}$, and a heterogeneous and divisible good X, which we refer to as the cake. From now on, our index set is $I = \{1, 2 \ldots k\}$. Each player P_i has a non-negative *measure function* μ_i that is a finitely additive measure so that $\mu_i(X) = 1$ for all $i \in I$. Besides this, each player P_i has a *demand* $d_i \in \mathbb{Z}^+$, representing that P_i demands $\frac{d_i}{\sum_{j \in I} d_j} \in]0, 1[$ part of the whole cake. The value of the whole cake is identical for all players, in particular it is the sum of all demands:

$$\forall i \in I: \quad \mu_i(X) = \sum_{j \in I} d_j.$$

Definition 3 *$\{X_i\}_{i \in I}$ is a* division *of X if $\bigcup_{i \in I} X_i = X$ and $\forall i \neq j : X_i \cap X_j = \emptyset$. We say that player P_i receives piece X_i. We talk about division with* equal shares *if $d_i = \frac{1}{k}$ for all $i \in I$. In the* unequal shares *setting, $d_i \in \mathbb{Z}^+$ and we define $n = \sum_{i \in I} d_i$.*

Definition 4 *We call division $\{X_i\}_{i \in I}$* fair *if*

$$\forall i \in I: \quad \mu_i(X_i) \geq \frac{d_i}{\sum_{j \in I} d_j}.$$

If the above inequality is fulfilled with strict inequality for all players, then the division is said to be strong fair.

Definition 5 *The* number of cuts *in an algorithm is the number of decisions that are made about cuts until termination. Let us denote the number of cuts for a k-player algorithm by $D(k)$.*

Notice that the last definition implies that choosing sides and calculating any other parameter than the value of a piece are not counted as cuts. We now highlight the number of cuts in two milestone algorithms in fair division with equal and unequal shares.

Theorem 6 (Evan and Paz [3]) *The number of cuts in the k-player equal-share Divide and Conquer Algorithm is*

$$D(k) = k \cdot \lceil \log_2 k \rceil - 2^{\lceil \log_2 k \rceil} + 1.$$

Theorem 7 (Robertson and Webb [4]) *The number of cuts in the k-player unequal-share Algorithm for More Than Two Players using the Cut Near-Halves Algorithm as a subroutine is*

$$D(k) = \sum_{i=2}^{k} \left[(i-1) \cdot \left\lceil \log_2 \left(\sum_{j=1}^{i} d_j \right) \right\rceil \right].$$

In particular, the 2-player Cut Near-Halves version requires $\lceil \log_2(d_1 + d_2) \rceil$ *cuts at most.*

We remark that the recursive formula in Theorem 7 can also be written in closed form as $O(k^2 \log n)$. Having established formal definitions and the most important theorems we are now ready to present our new algorithms.

3 Happiness in Unity Algorithm

First, we investigate the case of equal shares and assume that every player has a measure function that is absolutely continuous with respect to the Lebesgue measure. Our 'Happiness in Unity Algorithm' eliminates the role of the passive player from the Divide and Conquer Algorithm, and replaces it with a neutral referee, who is outside of the players' set. This modification gives every player the same chance to receive a larger piece than their fair share of $\frac{1}{k}$. Moreover, our algorithm computes a strong fair division if everyone decides to cut the cake in different places in the first round. At the end of this section, we show that our method requires asymptotically the same number of cuts as the Divide and Conquer algorithm.

Happiness in Unity Algorithm

If $k = 1$, then the sole player receives the whole cake.
If $k \geq 2$, then we distinguish two cases and introduce the role of a *referee*. She is a person outside of the game who evaluates each piece according to the Lebesgue-measure.

Case 1. In the first case $k = 2\ell$ for some $\ell \in \mathbb{Z}$. All players mark the cake X by parallel cuts in the ratio $1 : 1$. The referee cuts the cake between the ℓ-th mark and $\ell + 1$-th mark counted from left. The piece between those two cuts is halved with respect to the Lebesgue-measure. The ℓ players whose mark falls within the left side of referee's cut will perform the algorithm for $k = \ell$ player on that piece. The remaining ℓ players will perform such an algorithm on the right side.

Case 2. In the second case $k = 2\ell + 1$ for some $\ell \in \mathbb{Z}$. All players mark the cake X by parallel cuts in the ratio $\ell : \ell + 1$. The same happens as in the earlier case, except that ℓ player share the left side of the case, while the remaining $\ell + 1$ players share the right side.

Example 8 *Players Alice, Bob, Charlie and David are dividing the cake. Each player marks the cake with a parallel line where they would divide it into halves, see Figure 3. After this, the referee will cut the cake between the two middle marks to two equal pieces, with respect to her Lebesgue-measure. With this step we reduce the 4-player problem to two 2-player problems.*

Theorem 9 *If a player marks the cake at a different place than all others in the first round, then the Happiness in Unity Algorithm guarantees her a piece strictly larger than* $\frac{1}{k}$ *of the cake. In particular, if all players mark the cake at different places in the first round, then the Happiness in Unity Algorithm computes a strong fair division.*

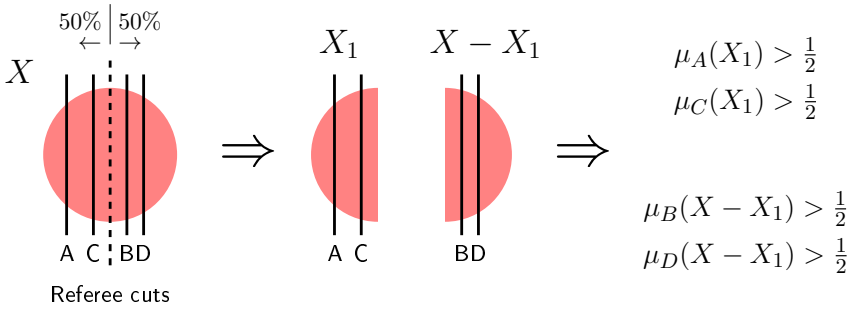


Figure 3: A reduction step in the Happiness in Unity Algorithm for an even number of players

Theorem 10 *The number of cuts in the k-player Happiness in Unity Algorithm is*

$$B(k) = k \cdot \lceil \log_2 k \rceil - 2^{\lceil \log_2 k \rceil} + k$$

With this number of cuts, Happiness in Unity Algorithm is no different from the optimal Divide and Conquer Algorithm in order of magnitude, see Theorem 6.

4 Joint-Stock Company Splitting Algorithm

We turn to the case of unequal shares now, where each player P_i has a demand $d_i \in \mathbb{Z}^+$. We change our terminology the following way. The cake is represented by a joint-stock company C, players are called share holders (P_i holds d_i shares) and the company is divided into $n = \sum_{i \in I} d_i$ shares. Our goal is twofold. On one hand, we want to split the joint-stock company into several smaller joint-stock companies $C^1, C^2, \ldots, C^m$ of $n^1, n^2, \ldots, n^m$ shares, respectively in such a way that $n = \sum_{j=1}^{m} n^j$. After such a split, a company C^j is called *profitable* for P_i if a share of C^j represents for P_i at least as much value as a share of C. On the other hand, we want to assign each company C^j to exactly one share holder such that each P_i holds exactly d_i shares in total and all these shares are in profitable companies.

We propose a k-player generalization of the Cut Near-Halves Algorithm to achieve the above goal. It is based on the following Joint-Stock Company Splitting Algorithm that splits a joint-stock company into two smaller companies and lets share holders swap their shares in a profitable way. We invoke this subroutine on each company that has more than one owner. So our method terminates when all companies have one exclusive owner.

In what follows, we describe the Joint-Stock Company Splitting Algorithm. The input of the algorithm is a k-tuple $(d_1, \ldots, d_k)$ describing the number of shares in company C of share holders $P_1, \ldots, P_k$ where $k \geq 2$. The output is a split of C into two companies C^1 and C^2 together with two k-tuples $(d_1^1, \ldots, d_k^1)$ and $(d_1^2, \ldots, d_k^2)$ describing the number of shares of the share holders in each of the companies such that $d_i^1 + d_i^2 = d_i$ for each $1 \leq i \leq k$. The algorithm starts with share holder P_k cutting the company either into *near-halves* or into *exact-halves*, depending on the value of $\mathcal{H}$ defined below.

Definition 11 *Let p be the number of odd numbers among $d_1, d_2, \ldots, d_{k-1}$, and let $B = \lceil \frac{p}{2} \rceil + \sum_{i=1}^{k-1} \lfloor \frac{d_i}{2} \rfloor$. We define*

$$\mathcal{H} = \begin{cases} 2 \cdot B - 1 & \text{if } \sum_{i=1}^{k-1} d_i \equiv 1 \ (mod \quad 2), \\ 2 \cdot B & \text{if } \sum_{i=1}^{k-1} d_i \equiv 0 \ (mod \quad 2). \end{cases}$$

Joint-Stock Company Splitting Algorithm

First calculate $\mathcal{H}$.

Case 1: $d_k \geq \mathcal{H}$. The greatest share holder P_k splits the company into two companies C^1 and C^2 with shares $\lceil \frac{n}{2} \rceil$ and $\lfloor \frac{n}{2} \rfloor$, respectively according to the **near-halves** principle. Out of C^1 and C^2, each share holder P_i picks a *more profitable* one, in which one share is worth more according to P_i. Each share holder P_i (for $1 \leq i < k$) receives $\lceil \frac{d_i}{2} \rceil$ shares from the more profitable company and $\lfloor \frac{d_i}{2} \rfloor$ shares from the less profitable one. No share holder is hurt by this distribution. After this, share holder P_k receives the unclaimed shares in companies C^1 and C^2. As P_k has split C, she does not get hurt. Now share holders start to *exchange shares*. This means that if some P_i and P_j label different companies as more profitable then (up to their shares) they exchange less profitable shares to more profitable ones. This exchange goes on till all the share holders (with a possible exception of P_k) have shares in exactly one of the companies C^1 and C^2.

In this case, we have either reduced the k-player problem to two problems with less share holders and shares, or we have decreased d_k, in case P_k exclusively owns C^1 or C^2.

Case 2: $d_k < \mathcal{H}$. We use the **exact-halves** principle, as the greatest share holder P_k divides C into C^1 and C^2 of equal value, each having n shares. Each share holder P_i gets d_i shares in both companies. Each share holder P_i (for $1 \leq i < k$) picks her more profitable company and starts to exchange shares as in Case 1. This goes on until no more profitable exchange is possible.

In this case, we have reduced the k-player problem into two, at most $(k-1)$-player problems.

Example 12 *Alice, Bob and Charlie divide the Fruit Company in ratio 3:28:7. The cutter will be Bob since he demands the biggest portion of the company. As $n = 38$, $d_k = 28$ and $\mathcal{H} = 10$, Bob splits the company into two equal value parts, the Apple and the Banana Company both of 19 shares. After the split, Alice and Charlie choose the more profitable part, see Figure 4a. So Bob receives 13 Apple shares and 15 Banana shares. First Alice and Bob and then Charlie and Bob exchange shares. This way, Bob receives all the Banana shares. So we have reduced Bob's demand and continue with dividing the 19-share Apple company between Alice, Bob and Charlie in ratio 3:9:7. Bob is still the one who splits, but this time $\mathcal{H} = 10 > 9 = d_k$ so Bob cuts exact-halves. Now Alice gets 3, Bob gets 9 and Charlie gets 7 shares in both parts, see Figure 4b. After share-exchange, we reduce the problem to two 2-player problems, and from now on, we follow the Cut Near-Halves Algorithm for both companies.*

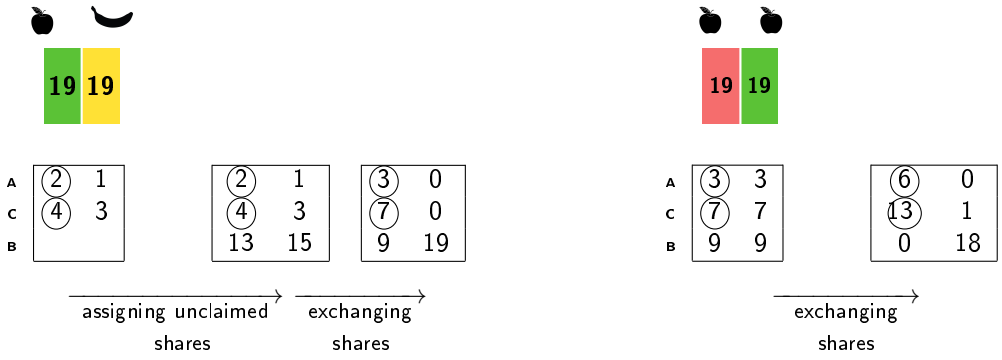
It is interesting to observe that for $k = 2$ share holders, in the Joint-Stock Company Splitting Algorithm, Case 2 cannot occur. Hence, in this case, our Joint-Stock Company Algorithm reduces to the Cut Near-Halves Algorithm. As an illustration, the reader might find useful to compare Figures 2 and 5.

Theorem 13 *If $d_1 < d_2$ and $(d_1, d_2) = 1$, then the number of cuts used in the 2-player Joint-Stock Company Splitting Algorithm in ratio $d_1 : d_2$ is*

$$D(n) = \lceil \log_2(d_1 + d_2) \rceil$$

Theorem 14 *If $2 \leq k$, $3 \leq n$ and $k < n$, then the number of cuts used for k-player Joint-Stock Company Splitting Algorithm on a cake of total value n can be estimated from above as:*

$$C(n, k) \leq (2^{k-1} - 1) \cdot \lceil \log_2 n \rceil + 1$$



(a) The first round in ratio 3:28:7

(b) The second round in ratio 3:9:7

Figure 4: The reduction steps in the 3-player Joint-Stock Company Splitting Algorithm in ratio 3:28:7

ratios		3 : 8		3 : 2		1 : 2		1 : 1	
companies		**5**	**6**	**2**	**3**	**1**	**2**	**1**	**1**
assigning unclaimed shares	AA	(2)	1	•	•	0	(1)	•	•
	BB	•	•	1	(1)	•	•	0	(1)
	AA	(2)	1	1	2	0	(1)	1	0
	BB	3	5	1	(1)	1	1	0	(1)
exchanging	AA	(3)	0	2	1				
	BB	2	6	0	(2)				
allocated companies	AA			{2}		{2}		{2},{1}	
	BB	{6}		{6}		{6},{1}		{6},{1},{1}	

Figure 5: A 2-player Joint-Stock Company Splitting Algorithm with ratio 3:8

5 Future directions

The number of cuts in the k-player Joint-Stock Company Splitting Algorithm remains clearly below the currently known best bound for a k-player fair division algorithm with unequal shares [4]. We remark though that our calculation might be tightened, moreover, the algorithm itself can potentially be modified so that the number of cuts decreases.

References

[1] L. E. Dubins and E. H. Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1):1–17, 1961.

[2] J. Edmonds and K. Pruhs. Cake cutting really is not a piece of cake. *ACM Transactions on Algorithms (TALG)*, 7(4):51, 2011.

[3] S. Even and A. Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7(3):285–296, 1984.

[4] J. Robertson and W. Webb. Cake-cutting algorithms: Be fair if you can. 1998.

[5] E. Romsics. Hogyan osszunk el igazságosan egy tortát sokfelé. Report, Budapest University of Technology and Economics, 2016.

[6] E. Romsics. Osztozkodási algoritmusok – Igazságos tortaosztás. BSc Thesis, Budapest University of Technology and Economics, 2016.

[7] G. J. Woeginger and J. Sgall. On the complexity of cake cutting. *Discrete Optimization*, 4(2):213–220, 2007.

The importance of having feedback – an application of matroid union in network analysis

Csongor Gy. Csehi[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
H-1521 Budapest, P.O.B. 91, Hungary
cscsgy@cs.bme.hu

András Recski[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
H-1521 Budapest, P.O.B. 91, Hungary
recski@cs.bme.hu

Abstract: The 10th meeting during the history of the Japanese-Hungarian symposium on discrete mathematics and its applications is a good opportunity to recollect memories of our earliest cooperations. We started joint research with Professor Masao Iri and his younger colleagues more than 40 years ago. Electric engineers often asked us: "Why do you use such abstract concepts as matroids instead of graphs for practical engineering problems? Could you show us problems which were untractable with good, old-fashioned graphs only?" In this talk we recollect one of our answers we had given at that time and mention some more recent results along the line of that area. Linear networks composed of 2-terminal devices are described by graphs. If the network contains multiterminal devices as well then the structure describing its independence properties will be a (not necessarily graphic) matroid. Using some recent results for characterizing graphicity of the union of matroids we show that in case of a single control the matroid will be graphic if and only if there is no feedback in the network.

Keywords: matroid theory, linear network, multiterminal devices

1 Introduction

Electric network analysis was the first real application of graph theory, almost 170 years ago. The laws of Kirchhoff [1] related the voltages and the currents of the devices to the circuits and cut sets, respectively, of the graph of the interconnection.

These classical results can be applied if the network consists of 2-terminal devices only. If the multiterminal devices are modeled by controlled sources then the interconnection can still be described by a graph but, due to the controls among the edges, the independence properties of the network graph will not properly describe the independences among the voltages or among the currents. Since the network is linear, it can be described by a single matrix but the column space matroid of this matrix will rarely be graphic.

The matroid operation union (also called sum) turned out to be the appropriate tool to describe the effect of control, as found independently by [2], [3] and [4]. However, the subset of graphic matroids is not closed with respect to union, in fact, the union of two graphic matroids is often outside the more general subset of binary matroids.

The fundamental results of [5] and [6] characterize those graphic matroids whose union is the free matroid (the cycle matroid of a tree). If the union of several copies of the same graphic matroid is considered then one can decide if this union is graphic [7] but the question is still open for general addends. A possible approach is to fix a graph G_0 or its cycle matroid $M_0 = M(G_0)$ and study those graphs G where the

[1]Research is supported by the grant # OTKA 108947 of the Hungarian National Research, Development and Innovation Office (NKFIH).

union of $M(G)$ and M_0 is graphic. If M_0 consists of loops only or it contains bridges then the problem is trivial hence the first interesting question was if G_0 consists of a circuit of length two (two parallel edges) and any number of loops. In the language of electric network analysis this corresponds to the linear active networks composed of 2-terminal passive devices plus a single current controlled current source. This case has been solved in [8] – mathematically it was a Kuratowski-type characterization of G which had a physical interpretation as the lack of feedback, see Theorems 1 through 4 below.
The results of [8] have recently been generalized for the case if G_0 consists of either n series or n parallel edges in addition to the loops, see [9] for $n = 3$ and [10] for any n. In the present paper we study the interpretation of the structure of G_0 in terms of controlled sources, and formulate the mathematical meaning of these recent results in the language of electric network analysis.

2 Former Results

Throughout, we use the notation of [11]. Suppose that a network is composed of 2-terminal devices and current controlled current sources (CCCS). The graph of the network is defined in the usual way (each CCCS corresponds to a pair of edges), and we assign orientation to each edge arbitrarily. There are several equations among the currents of the devices, some of them are the Kirchhoff Current Laws, describing the topology of the network, some others describe the controls. In what follows, we shall refer to these sets of equations as the graphic and the algebraic sets of equations, respectively. For example, Figure 1 shows a network on the left and its graph on the right, the set of the graphic equations consists of

$$i_1 + i_2 + i_3 = 0$$

$$i_3 - i_4 - i_5 = 0$$

(and any linear combinations of them), while there is a single algebraic equation $i_5 = c \cdot i_2$.

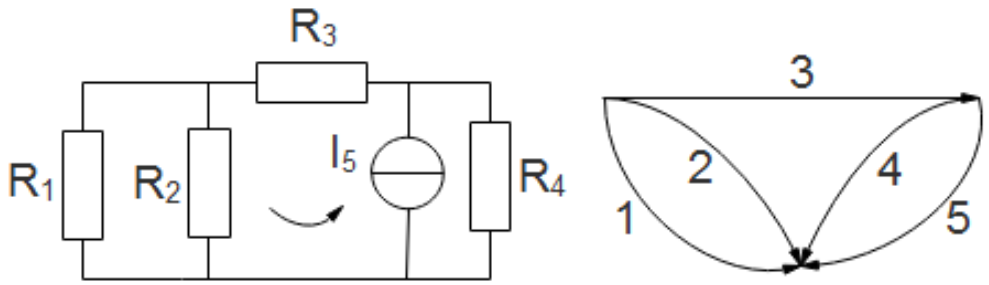


Figure 1: A network and the corresponding graph

Hence there are three linear equations referring to the five currents and these equations can be summarized by the coefficient matrix

$$M_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & c & 0 & 0 & -1 \end{pmatrix}$$

In contrast, the network of Figure 2 has a different kind of control, namely $i_5 = c \cdot i_3$, hence our matrix will be

$$M_2 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & c & 0 & -1 \end{pmatrix}$$

The column space matroid of M_2 is graphic, see the left hand side of Figure 3, while that of M_1 is not –

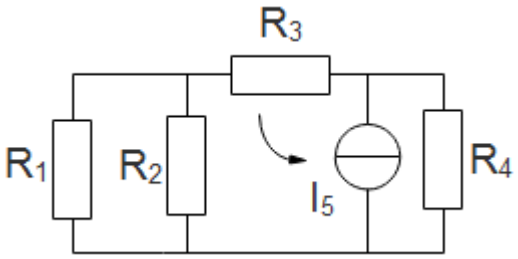


Figure 2: A network, similar to that of Figure 1 but with a different kind of control

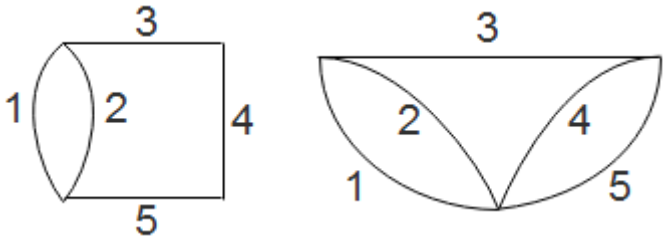


Figure 3: A graph representing M_2 and the graph representing the interconnection of both networks

no one can draw a graph with four vertices and five edges so that $\{1, 3, 4\}$ is a circuit and any other set of three edges forms a spanning tree.

The above examples illustrate the necessity of the condition in the following theorem:

Theorem 1 *[8] Let G_0 consist of a circuit of length two (two parallel edges a, b) and any number of loops. Let M_0 denote the cycle matroid $M(G_0)$. Let G be an arbitrary graph on the same edge-set. Then the union of $M(G)$ and M_0 is graphic if and only if G does not contain any subgraph isomorphic to the graph of Figure 4 or to its subdivision, with a and b in the indicated positions.*

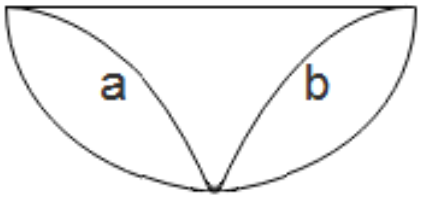


Figure 4: The graph whose existence characterizes the presence of feedback

If a network is composed from 2-terminal devices and of a single CCCS (whose edges will play the role of a and b) then the existence of the subgraph of Figure 4 or its subdivision (with a and b in the requested positions) means the presence of a feedback F, no matter what kind of subnetworks N_1, N_2 are interconnected, see Figure 5. Hence the above theorem can be reformulated as follows:

Theorem 2 *[8] Suppose that a network is composed of 2-terminal devices and of a single current controlled current source. The independence structure describing the currents of the devices is graphic if and only if there is no feedback in the network.*

The graph G in Theorem 1 was arbitrary. In network theory applications we may always suppose that the underlying graph of the electric network is connected, in fact, even 2-connected if there is no control

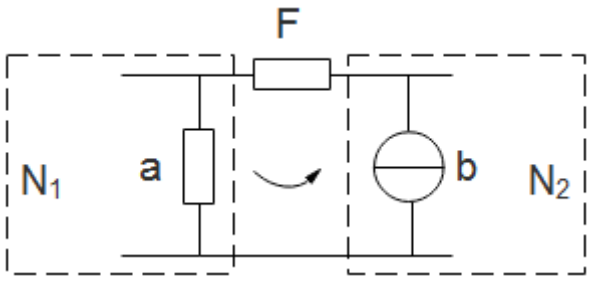


Figure 5: Feedback in a general network

in the network. Moreover, if a subgraph is connected along two points to the rest of the graph and none of the edges of this subgraph is a controlling or a controlled element then the whole subgraph can be replaced by a single edge. Using these replacements if applicable, we obtain the reduced graph of the network. For a more formal description of this matroid theoretical reduction see Section 2 of [10].
In view of this, *feedback* is formally defined as the presence of at least one circuit in the complement of $\{a, b\}$ in the reduced network graph. Then one can reformulate Theorem 1 as follows:

Theorem 3 *Suppose that the reduced graph of the network is 2-connected and a, b are two non-serial edges. Then there is a subgraph isomorphic to Figure 4 or its subdivision, with a and b in the specified positions, if and only if the complement of $\{a, b\}$ in the reduced network graph contains at least one circuit.*

In the next section we shall refer to the negative of this reformulation:

Theorem 4 *Suppose that a network is composed of 2-terminal devices and of a single current controlled current source involving the edges a, b. We may suppose without loss of generality that the reduced graph of the network is 2-connected and a, b are two non-serial edges. Then the independence structure describing the currents of the devices is graphic if and only if there is no feedback in the network, that is, if the complement of $\{a, b\}$ in the reduced network graph is circuit-free.*

In what follows we shall generalize Theorems 2 and 4 for more general types of control. Recall that in case of a CCCS the current of a single source is controlled by the current of a single resistor. We have found analogous results if only one of these restrictions remains.

3 New Results

3.1 Several Controlled Sources and a Single Controlling Element

Suppose that the current of a single resistor R_0 controls several current sources $I_1, I_2, ..., I_k$ as described by the respective equations $i_j = c_j \cdot i_0$ for every $j = 1, 2, ..., k$. We may suppose that the set S of the corresponding edges $e_0, e_1, e_2, ..., e_k$ does not contain any cut-set in the graph of the network, since otherwise there were an additional equation $\Sigma i_j = 0$ among some of these currents, which, together with the control equations $i_j = c_j \cdot i_0$, would lead to a singular network.
Since there are k controls in the network, the above definition of the feedback is modified as the presence of at least one circuit in the complement of the set S in the reduced network graph.

Theorem 5 *Suppose that a network is composed of 2-terminal devices and the current of a resistor R_0 controls several current sources $I_1, I_2, ..., I_k$ as described by the respective equations $i_j = c_j \cdot i_0$ for every $j = 1, 2, ..., k$ (where the control constants $c_1, c_2, ..., c_k$ are generic parameters, that is, they are algebraically independent over the field of the rational numbers). We may suppose without loss of generality that the above set S is cut set free. Then the independence structure describing the currents of the devices is graphic if and only if there is no feedback in the network, that is, if the complement of S in the reduced network graph is circuit-free.*

PROOF: The system of equations $i_j = c_j \cdot i_0$ for every $j = 1, 2, ..., k$ leads to an algebraic submatrix representing a matroid M_1 which consists of loops and a single circuit of length $k+1$. Let M_2 denote the matroid, represented by the graph of the interconnection. Proposition 14 of [10] states that the union of the reduced matroids M_1' and M_2' is graphic if and only if either S contains a cut-set or $M_2' \setminus S$ is the free matroid. Since the former case is excluded, the reduced network graph without the edges in S must be circuit-free. □

3.2 Several Controlling Elements and a Single Controlled Source

Suppose that a single current source i_0 is controlled by the current of several resistors $R_1, R_2, ..., R_k$ as described by the equation $i_0 = \Sigma(c_j \cdot i_j)$ where the summation is for every $j = 1, 2, ..., k$. We may suppose without loss of generality that the network graph is either 2-connected or the set S of the corresponding edges $e_0, e_1, e_2, ..., e_k$ has at least one edge from each 2-connected component.
Since there is a single control involving $k+1$ elements in the network, the above definition of the feedback is modified as the presence of at least one circuit in the complement of any two-element subset of the set S in the reduced network graph.

Theorem 6 *Suppose that a single current source i_0 is controlled by the current of several resistors $R_1, R_2, ..., R_k$ as described by the equation $i_0 = \Sigma(c_j \cdot i_j)$ where the summation is for every $j = 1, 2, ..., k$. Like in Theorem 5, we suppose that the control constants $c_1, c_2, ..., c_k$ are generic parameters, that is, they are algebraically independent over the field of the rational numbers. We may suppose without loss of generality that the network graph is either 2-connected or the set S of the corresponding edges $e_0, e_1, e_2, ..., e_k$ has at least one edge from each 2-connected component. Then the independence structure describing the currents of the devices is graphic if and only if there is no feedback in the network, that is, if the complement of the edge set $\{a, b\}$ is circuit-free for any two non-serial edges a, b of S in the same 2-connected component of the reduced network graph.*

PROOF: The equation $i_0 = \Sigma(c_j \cdot i_j)$ leads to an algebraic submatrix representing a matroid M_1 which consists of loops and $k+1$ parallel edges. Proposition 22 of [10] states that the union of the reduced matroids M_1' and M_2' is graphic if and only if no 2-connected component of the reduced network graph G has two non-serial edges a, b so that $G - \{a, b\}$ contains a circuit. This is clearly equivalent to the condition of Theorem 6. □

4 Examples and remarks

Example 7 *Consider the network of Figure 6 where $i_0 = c_1 \cdot i_1 + c_2 \cdot i_2$. The graph of the network is given in Figure 7. The coefficient matrix for the system of equations for the currents of the elements will be*

$$\begin{pmatrix} -1 & c_1 & c_2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

The matroid represented by the columns of this matrix has six elements and rank four. This matroid is non-graphic – if we contract elements 4 and 5 then the resulting minor is the rank 2 uniform matroid on the set $\{0, 1, 2, 3\}$ which is known not to be binary, let alone graphic. Based on Theorem 6 one could reach the same conclusion: The elements 0 and 1 are non-serial edges in the same 2-connected component of the graph of Figure 7, still the complement of the set $\{0, 1\}$ contains a circuit, namely $\{2, 5\}$.

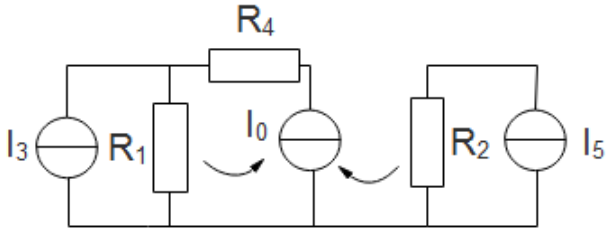


Figure 6: The network of Example 7

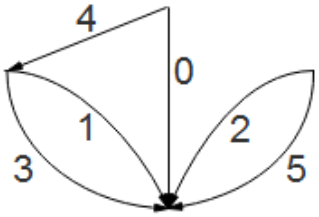


Figure 7: The graph of the network of Example 7

Example 8 *The network of Figure 8 illustrates Theorem 5. Let the controls be* $i_1 = c_1 \cdot i_0$ *and* $i_2 = c_2 \cdot i_0$. *The graph of the network is given in Figure 9 and the coefficient matrix for the system of equations for the currents of the elements will be*

$$\begin{pmatrix} c_1 & -1 & 0 & 0 & 0 & 0 & 0 \\ c_2 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 \end{pmatrix}$$

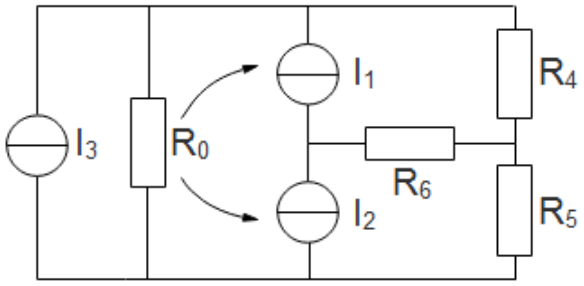


Figure 8: The network of Example 8

The corresponding matroid has seven elements and rank five. One can see that it is non-graphic – if we contract elements $0, 2$ and 6, the resulting minor is the rank 2 uniform matroid on the set of the remaining elements. Based on Theorem 5 one could reach the same conclusion: If we delete the edges of the set $S = \{0, 1, 2\}$ from the graph of Figure 9, the remaining graph contains a circuit, namely $\{3, 4, 5\}$.

Remark 9 *Our new results either released the requirement that a single source is controlled only, or that the source is controlled by a single element only. However, if there are several controlled sources controlled by distinct elements then the describing matroid can be graphic again, as shown by the following example.*

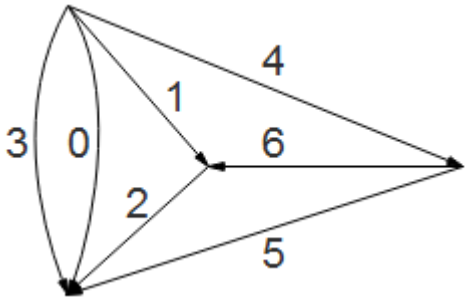


Figure 9: The graph of the network of Example 8

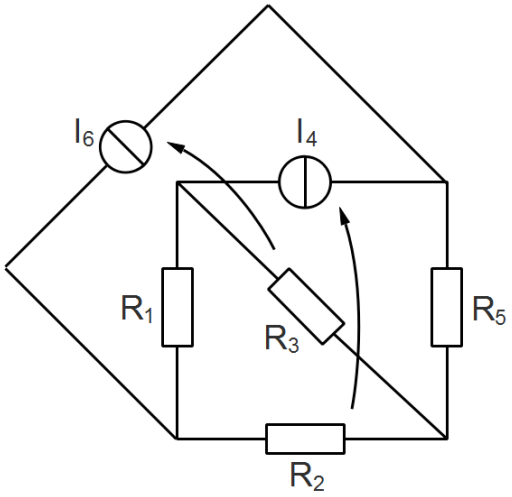


Figure 10: The network of Example 10

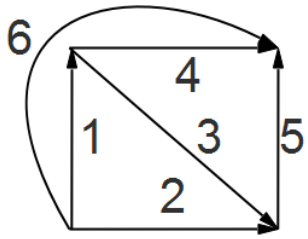


Figure 11: The graph of the network of Example 10

Example 10 *Consider the network of Figure 10 where* $i_6 = c_1 \cdot i_3$ *and* $i_4 = c_2 \cdot i_2$. *The graph of the network is given in Figure 11. The coefficient matrix for the system of equations for the currents of the elements will be*

$$\begin{pmatrix} 1 & 0 & -1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & c_1 & 0 & 0 & -1 \\ 0 & c_2 & 0 & -1 & 0 & 0 \end{pmatrix}$$

One can easily check that any five of the six columns are linearly independent, hence the corresponding matroid is graphic (the cycle matroid of a single circuit) for most values of c_1 and c_2 (see below).

Remark 11 *Results applying matroid union for engineering applications frequently require a genericity-type condition like the one we had in Theorems 5 and 6 concerning the control constants $c_1, c_2..., c_k$. The basic reason of this has been discovered by Edmonds [12] during his study about the relation between rank and term rank of the matrices. If such an assumption is missing, the statement might be wrong.*

Example 12 *Suppose that $c_1 = -1$ in Example 7. Then the set $\{0, 1, 4\}$ will become a circuit and the matroid will be graphic (a circuit formed by $\{0, 1, 4\}$ and another formed by $\{1, 2, 3, 5\}$, sharing a common edge). Physically, it corresponds to a singular network: The relation $c_1 = -1$ leads to a control equation $i_0 = -i_1 + c_2 \cdot i_2$; hence the Kirchhoff equation $i_3 = -(i_1 + i_0)$ would lead to a relation $i_3 = c_2 \cdot i_5$ between two independent current sources.*

Example 13 *Consider the network of Figure 10. If $c_1 = c_2 = -1$ then the rank of the matroid will decrease to 4 with every four-tuple except $\{1, 2, 4, 5\}$ and $\{1, 3, 5, 6\}$ being a base. If we contract, say, elements 2 and 3, the resulting minor is the rank 2 uniform matroid on the set of the remaining elements, showing that it is not graphic.*

References

[1] G. Kirchhoff, Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird, *Ann. Phys. Chem.* **72** (1847) 497–508.

[2] A. Recski, On partitional matroids with applications, *Colloq. Math. Sot. J. Bolyai* **10** (1973) 1169–1179.

[3] M. Iri and N. Tomizawa, A unifying approach to fundamental problems in network theory by means of matroids, *Trans. Inst. Electron. & Commun. Eng. Jpn.* **57A, 8** (1975) 35–41.

[4] J. Narayanan, Theory of Matroids and Network Analysis, *Ph.D. Thesis in Elect. Engin., Indian Institute of Tech., Bombay, India* (1974)

[5] C. St. J. A. Nash-Williams, Decomposition of finite graphs into forests, *Journal of the London Mathematical Society* **39 (1)** (1964) 12.

[6] J. Edmonds, Minimum partition of a matroid into independent subsets, *J. Res. Nat. Bur. Stand* **69B** (1965) 67–72.

[7] L. Lovász and A. Recski, On the sum of matroids, *Acta Math. Acad. Sci. Hungar.* **24** (1973) 329–333.

[8] A. Recski, On the sum of matroids II, *Proc. 5th British Combinatorial Conf. Aberdeen* (1975) 515–520.

[9] Cs. Gy. Csehi, Matroidelméleti vizsgálatok és alkalmazásaik, *Master's Thesis, Budapest University of Technology and Economics,* (2012)

[10] Cs. Gy. Csehi and A. Recski, The graphicity of the union of graphic matroids, *European Journal of Combinatorics* **50** (2015) 38–47.

[11] A. Recski, Matroid Theory and its Applications in Electric Network Theory and in Statics, *Springer, Berlin* (1989)

[12] J. Edmonds, Systems of distinct representatives and linear algebra, *J. Res. Nat. Bur. Standards 71B* **71B** (1967) 241–245.

Limit theory of discrete mathematics problems

Endre Csóka[1]

Alfréd Rényi Institute of Mathematics,
Budapest, Hungary
csokaendre@gmail.com

Abstract: We show a general problem-solving tool called limit theory. This is an advanced version of asymptotic analysis of discrete problems when some finite parameter tends to infinity. We will apply it on three closely related problems.

Alpern's Caching Game (for $k = j = 2$) is defined as follows. The hider caches 2 nuts into one or two of n potential holes by digging at most 1 depth in total. The goal of the searcher is to find both nuts in a limited time h, otherwise the hider wins. We will show that if h and n/h are large enough, then very counterintuitively, any optimal hiding strategy should dig less than 1 in total, with positive probability. We will prove it by defining and analyzing a limit problem. Then we will partially solve the entire problem, and we will also have significant progress with the Manickam–Miklós–Singhi Conjecture and the Kikuta–Ruckle Conjecture.

Keywords: combinatorics, discrete mathematics, asymptotic analysis

1 Introduction

Limit theory techniques were already used in different areas. Statistical physics is essentially just limit theory in physics. In mathematics, probably Fürstenberg (1977, [6]) was the first to use limit theory techniques, for reproving Szemerédi's Theorem. Lovász and Szegedy (2006, [9]) used this technique by introducing limit graphs called graphons. This also motivated the limit theory of many different discrete structures. However, in all these cases, limit theory was used only for special purposes. In this paper, we will show that this is indeed a general problem-solving tool.

We can describe the technique in an abstract way as follows. For a problem $P(\underline{x})$ with some real (often integral) parameters $\underline{x}$, we call another problem L a limit problem of P if (an important parameter of) the solution of L is the limit of (the parameter of) a sequence of solutions of $P(\underline{x}^{(i)})$ for some parameter vectors $\underline{x}^{(1)}, \underline{x}^{(2)}, \underline{x}^{(3)}, \ldots$. Finding a good limit problem can be difficult but very useful. We will show it through several examples.

Our most convincing but most difficult application is about Alpern's Caching Game (Section 2). We will find and analyze multiple limit problems, and we will get to some very interesting and highly counterintuitive results. Then we will extend the Manickam–Miklós–Singhi (MMS) Conjecture (Section 3) using a simple but nontrivial limit problem. Finally, we will consider the Kikuta–Ruckle (KR) Problem (Section 4), which is a generalization of the MMS Problem. We will understand and specify the KR Conjecture in a highly nontrivial way, using some limit problems. About Alpern's Caching Game and the KR Problem, the solutions depending on the parameters looked chaotic, but our techniques will reveal the structures in them.

2 Alpern's Caching Game

Definition 2.1. *Alpern's Caching Game $G(k, j, n, h)$ is defined as the following 2-player game between the hider and the searcher. There are $n \in N$ (potential) holes and $k \in N$ nuts. The hider places (caches)*

[1]The research was supported by Marie Skłodowska-Curie grant 750857

each of k nuts into one of the holes in a positive depth, so that if we take the depth of the deepest nut in each non-empty hole, then their sum must be at most 1. The searcher cannot observe anything about the placement, but he can dig the hole at most depth $h \in \mathbb{R}^+$ in total. A nut is found if the searcher dug at least as much in that hole as the depth of the nut. The searcher can choose an adaptive digging strategy, continuously observing what and where he already found during the digging.[1] *The searcher wins if he finds at least $j \in N$ out of the k nuts. Otherwise the hider wins.*

This problem was introduced by Alpern, Fokkink, Lidbetter and Clayton in [2], a summary about the results and related questions of Alpern's Caching Game can be found in the survey book by Alpern, Fokkink, Leszek, Lindelauf [1]. More recent results are presented in [4].

This is a 2-player 0-sum game, therefore, there is a **value of the game** $v = v(k, j, n, h)$ with the following properties. The hider has a strategy so that for any strategy of the searcher, the searcher wins with probability at most v. Similarly, the searcher has a strategy so that he wins with probability at least v against any strategy of the hider. These are called optimal strategies.

This problem is solved for $k = j = 2$, $n \leq 4$. [2, 4] The solution for $k = j = 2$, $n = 4$ is the following stepfunction.

h	$[0,1)$	$[1,\frac{3}{2})$	$[\frac{3}{2},\frac{5}{3})$	$[\frac{5}{3},\frac{7}{4})$	$[\frac{7}{4},\frac{9}{5})$	$[\frac{9}{5},\frac{11}{6})$	$[\frac{11}{6},2)$	$[2,\frac{11}{5})$	$[\frac{11}{5},\frac{7}{3})$	$[\frac{7}{3},3)$	$[3,4)$	4
$v(2,2,4,h)$	0	$\frac{1}{10}$	$\frac{3}{20}$	$\frac{1}{5}$	$\frac{9}{40}$	$\frac{7}{30}$	$\frac{1}{4}$	$\frac{2}{5}$	$\frac{9}{20}$	$\frac{1}{2}$	$\frac{3}{4}$	1

The optimal strategies show an even more chaotic picture: almost all are completely different in the different intervals of h. Therefore, the solution even for $k = j = 2$ seemed to be chaotic and almost hopeless to characterize. However, using limit theory, we will show that it is not the case. First, we show a surprising property of the solution, and then we will partially solve the problem for $k = j = 2$. Some of the results will apply for $k = j > 2$ and $k = j + 1 > 2$.

Definition 2.2. *In Alpern's Caching Game, we say that a placement (or pure hiding strategy) of the nuts is **extremal** if the sum of the depths of the deepest nuts in the different holes is exactly 1. A (mixed) hiding strategy is called extremal if it is supported on extremal placements. The extremal version of the game XG means that the hider must use an extremal strategy.*

Let v_X always denote the same as v with the extremal version of the game.

Question 2.3. *Does the hider always have an extremal strategy which is optimal?*
Or (equivalently) does $v(k, j, n, h) = v_X(k, j, n, h)$ always hold?

The answer was believed to be clearly positive, some of the authors of the problem did not even realize that they did not have a proper proof of it (according to private communications). Moreover, there was a conjecture presented in [1] about a difficult recursive property of the optimal strategies of the hider, which was in accordance to the solved cases, but which implied the positive answer to Question 2.3.

In the beginning, the author of this paper was almost sure about the positive answer, too. But limit theory analysis pointed out the opposite. On the top of it, further analysis showed that for a large class of parameters, the optimal hiding strategy uses non-extremal placements with probability 1. The proof of our final results are presented in Subsection 2.3. However, the primary goal of this section is not showing the final results and the proofs like pulling a rabbit out of a hat, but we rather explain the idea which we believe to be a general problem-solving technique. The same technique will be used in the other two sections of this paper.

Define the limit of the game when the number of nuts to hide k and to find j are fixed, but the number of holes and the digging time $n, h \to \infty$, with an asymptotic ratio $n/h \to \lambda$.

[1] Strategy means mixed strategy. If we use the discrete sigma-algebra on the set of strategies, then there is nothing to add to the definition, but we cannot use continuous distributions about the hiding depths and we will have only approximately optimal strategies. Or we can use the Lebesgue sigma-algebra on the set of hiding strategies, but in this case, we need some measurability criteria for the searching strategy in order to define winning probabilities. These are irrelevant issues about our results, and we will omit these technical details throughout the section.

Definition 2.4. *The **limit game** $LG(k,j,\lambda)$ and its extremal version $XLG(k,j,\lambda)$ are defined as follows. The hider chooses a partitioning $a_1,a_2,...,a_{k'} \in \mathbb{Z}^+$ where $\sum a_i = k$, and he chooses values (depths) $y_1,y_2,...y_{k'} \in [0,1]$ with $\sum y_i \leq 1$ in LG, and $\sum y_i = 1$ in XLG. Then for k' independent uniform random numbers $x_1,x_2,...x_{k'} \in [0,\lambda]$, a_i number of nuts are placed at (x_i,y_i). The searcher observes nothing. Now the searcher should define a function $f_t(x) : [0,1] \times [0,\lambda] \to [0,1]$ which is monotone increasing in both parameters and $\int f_1(x)\ \mathrm{d}x \leq 1$. Then we evaluate f meaning that the searcher gets to know the smallest t^* so that $f_{t^*}(x_i) \geq y_i$ for some i. If there is no such nut position even for f_1, then the game ends. Otherwise the nuts at (x_i,y_i) are found by the searcher, he gets to know the position and the number of them, and we remove these nuts. Then the searcher can change his function in the parameter interval $t \in (t^*,1]$, and we re-evaluate f. The searcher wins if he finds at least j nuts in total.*

Note 2.5. *We can get an equivalent problem by assuming that $\int f_t(x)\ \mathrm{d}x = t$ for all $t \in [0,1]$. This will be convenient to assume when we are showing lower bounds on the value of the game. Also, we can omit the condition that f is monotone increasing in the second coordinate, which will be useful for upper bounds.*

The following theorem shows the reason why we call it a limit game. Denote the value of the limit game by $v(k,j,\lambda)$.

Theorem 2.6. *For any parameters $k,j,n,h \in \mathbb{Z}^+$,*

$$v(k,j,n,h) : v\Big(k,j,\frac{n}{h}\Big) \in \Big[\Big(\frac{h-j}{h}\Big)^j, 1\Big].$$

Therefore, if $n_i/h_i \to \lambda$ and $n \to \infty$, then $v(k,j,n_i,h_i) \to v(k,j,\lambda)$. The same applies for the extremal versions.

This theorem will be used when we will disprove Question 2.3 for the first time. But we will not need this for the final results. Therefore, we present just a sketch of proof of the theorem.

Sketch of proof. Notice first that both players can choose a uniform random permutation of the holes and apply his strategy on this permutation. If either of them does so, then whether the other player does it makes no difference in the expected result. Therefore, if we add to the rules that either or both players must use this randomization, then it does not change the value of the game, as both players can secure himself this expected score.

About the limit games LG and XLG, notice that if two holes are dug to the same depth, and nothing was found in them so far, then it does not matter which one the searcher continues digging. Therefore, we can assume that according to the random ordering of the holes, their depths remain monotone decreasing during the search, excluding holes in which we have already found a nut.

Now let us see why do the values of the discrete problems converge to the values of the limit problems.

On one hand, any strategy of the searcher in the discrete game G or XG can be applied in the limit game by choosing f_t in the interval $\left[\frac{i-1}{n},\frac{i}{n}\right)$ as the depth of the ith deepest hole after a total amount of digging t. This way the searcher can get at least the same score as in the discrete game.

On the other hand, a strategy of the searcher in LG (or XLG) can be applied in G (or XG) as follows. The searcher chooses a random ordering of the holes. Then he digs so as to have depth $f_t\left(\frac{i-1}{h-j}\right)$ in the ith hole, except that if a nut is found in a hole, then he digs that hole until depth 1. He does it for all $t \in [0,1]$, in increasing order. This way, the searcher can get at least $\left(\frac{h-j}{h}\right)^j$ times the score of the limit game LG (or XLG). □

Definition 2.7. *The **double limit game** $DLG(k,j)$ and its extremal version $XDLG(k,j)$ are defined as the limit game with $\lambda \to \infty$, as follows. The hider chooses k values $y_1,y_2,...y_k \in [0,1]$ where $\sum y_i \leq 1$ in DLG, and $\sum y_i = 1$ in $XDLG$. At the same time, the searcher defines a pure strategy of the limit game with $\lambda = \infty$. Then for each subset $Q = \{q_1,q_2,...,q_j\} \subset \{1,2,...,k\}$ with a vector of positive real numbers $x_{q_1},x_{q_2},...x_{q_j} \in \mathbb{R}^+$, the nuts are placed at (x_{q_i},y_{q_i}). The score of the searcher is the j-dimensional measure of the vectors $x_{q_1},x_{q_2},...x_{q_j}$ for which he wins by his strategy, summing up for all different Q. This is what the searcher aims to maximize and the hider aims to minimize, in expectation.*

Denote the value of $DLG(k,j)$ by $v(k,j)$.

Theorem 2.8. *Fix $k \geq j \geq 2$, and consider a sequence of pairs (n_i, h_i) so that $h_i \to \infty$ and $\frac{n_i}{h_i} \to \infty$. Then*

$$\left(\frac{n_i}{h_i}\right)^j \cdot v(k,j,n_i,h_i) \to v(k,j).$$

The same applies for the extremal versions.

Sketch of proof. The strategy of the searcher in LG can be applied in DLG. This shows one direction.

The other direction is a bit more technical. The first observation is that in G, if the hider puts more nuts in the same hole, and the searcher digs $\lfloor h \rfloor$ holes until depth 1, then this already provides him a score $\omega\big(\big(\frac{n_i}{h_i}\big)^{-j}\big)$. Therefore, in the optimal hiding strategy of the hider, the probability of such a placement tends to 0. Thus, the limit of the values does not change if we forbid such a placement in G.

The next observation is that the probability of finding more than j nuts is $o\big(\big(\frac{n_i}{h_i}\big)^{-j}\big)$. Therefore, the probability of finding j nuts is essentially the same as the expected number of j-element subsets of the nuts which would be found by the searcher if the other nuts had not been cached.

The optimal strategy of the searcher in DLG can be applied in LG with a large parameter λ, simply by restricting f to $[0,1] \times [0,\lambda]$. In DLG, if $x > \lambda$, then (by monotonicity) $f_1(x) < \frac{1}{\lambda}$ throughout the game. Therefore, this restricted strategy provides the same score unless if the depth of a nut is at most $\frac{1}{\lambda}$.

The following searching strategy is very efficient if the depth of a nut is at most $\frac{1}{\lambda}$. First, the searcher chooses $f_t(x) = \frac{1}{\lambda}$ if $x < \frac{t}{\lambda}$, and 0 otherwise. Then, after finding the first nut at (x_1, y_1), then he chooses $f_1(x) = \frac{1}{\lambda}$ if $x < x_1$, $f_1(x) = 1$ if $x \in [x_1,\ x_1 + 1 - \lambda x_1]$ and 0 otherwise. If we use this strategy with probability $O\big(\frac{1}{\lambda}\big)$ and the strategy f restricted to $[0,1] \times [0,\lambda]$ otherwise, then for all possible hiding strategy, this mixed searching strategy in LG will be (at least) almost as good as the original strategy in DLG.

The same argument works for the extremal games, as well. □

Consider any optimal strategy of the hider in $XDLG(2,2)$. This can be identified with the probability measure μ of the depth of a random nut.

Lemma 2.9. *In $XDLG(2,2)$, if the searcher with any optimal strategy finds a nut at depth $y \in \operatorname{supp}(\mu)$, then he almost always changes f so as to maximize the size of the interval $f_1^{-1}(1-y) = \big\{x \in [0,\infty) : f_1(x) = 1-y\big\}$.*
More precisely, if the hider chooses a placement randomly from μ, and the searcher plays optimally, then finding a nut and changing f not in the suggested way happens with probability 0.

Sketch of proof. Otherwise the searcher could improve his score against an optimal hiding strategy. □

Two pure searching strategies are called *equivalent* if they get the same score against any hiding strategies including the non-extremal ones. Two (mixed) searching strategies are equivalent if there exists a measure preserving bijection between the two distributions of pure strategies such that the corresponding pure strategies are equivalent except for a 0-measure set.

Lemma 2.10. *For any optimal strategy of the searcher in $XDLG(2,2)$, there is an equivalent one which starts with a function f satisfying that for all $(t,x) \in [0,1] \times [0,\infty)$,*

$$f_t(x) = 0 \quad \text{or} \quad \forall \varepsilon > 0 \colon \mu\big[f_t(x) - \varepsilon,\ f_t(x)\big] > 0. \tag{1}$$

(This is a little more restrictive than $f_t(x) \in \{0\} \cup \operatorname{supp}(\mu)$.)

Sketch of proof. Assume that $\int f_t(x)\ \mathrm{d}x = t$. Let

$$q_t(x) = \sup\Big\{y \leq f_t(x) \;\Big|\; (y = 0) \text{ or } \big(\forall \varepsilon > 0 : \mu[y-\varepsilon,\ y] > 0\big)\Big\}.$$

Clearly, f and q get the same score against the hiding strategy μ. Now, after some case analysis, we can get to the following conclusion. Either q is an equivalent searching strategy to f, or we can find an $\hat{f} \geq q$ which provides higher score than f against μ. □

Theorem 2.11. $v_X(2,2) < v(2,2)$.

Corollary 2.12. *Theorems 2.8 and 2.11 imply the existence of infinitely many counterexamples for Question 2.3.* □

Sketch of proof of Theorem 2.11. Assume by contradiction that $v_X(2,2) = v(2,2) = v$. Consider an optimal strategy S' of the searcher in DLG. This provides the expected score at least v against any strategy of the hider. Therefore, S' provides an expected score at least v against all extremal hiding strategies, therefore, S' is an optimal searching strategy in $XDLG$, as well. Lemma 2.10 says that there exists an S equivalent to S' satisfying (1). Consequently, S is an optimal searching strategy in DLG which satisfies (1). This can be represented by the first-round searching function f^S.

μ and S are optimal hiding and searching strategies in both DLG and $XDLG$, because they provide an expected score at least and at most v, respectively, against any searching strategy. According to Lemma 2.9, the strategy of the searcher is represented by the first function f he chooses.

Case 1. $\mathrm{supp}(\mu) = [0,1]$. (We conjecture this to be true.)

In $XDLG$, a randomization between the following two pure strategies of the searcher provides him a score at least $\sqrt{2}+1$.

- $f_t(x) = 1$ if $x < t$, otherwise 0. – This scores $\frac{1}{2y_1y_2}$.
- $f_t(x) = \frac{1}{2}$ if $x < 2t$, otherwise 0. – This scores $\frac{1}{y_2} + \frac{1}{2y_2^2}$, where $y_2 \geq y_1$.

Therefore,

$$v \geq \sqrt{2}+1. \tag{2}$$

Consider an arbitrary strategy of the searcher in $XDLG$. Let t^* denote the time point when $\left(\frac{4}{3}, \frac{1}{4}\right)$ is dug, and let s_1 and s_2 denote the length of holes which had been dug to the depth at least $\frac{1}{4}$ before or by t^*, respectively. Formally,

$$t^* = \sup_{t\in[0,1]} \left\{ f_t\Big(\frac{4}{3}\Big) < \frac{1}{4} \right\}, \qquad s_1 = \inf_{x\in\mathbb{R}} \left\{ \sup_{t\in[0,1]} \left\{ f_t(x) < \frac{1}{4} \right\} = t^* \right\}, \qquad s_2 = \sup_{x\in\mathbb{R}} \left\{ f_{t^*}(x) \geq \frac{1}{4} \right\}.$$

Now consider the score it provides against the hiding strategies $y_1 = \frac{1}{4} - \varepsilon$ and $y_2 = \frac{1}{4}$ when $\varepsilon \to 0+$. In the limit, the searcher can win only if one of the followings are satisfied.

- $\max(x_1, x_2) \leq s_1$ and $x_1, x_2 \leq \frac{4}{3}$
- $x_1 \in [s_1, s_2]$ and $x_2 \in \left[s_1,\ 2 - \frac{s_2}{2}\right]$

The total area of the set of these pairs $(x_1, x_2) \in [0,\infty)^2$ is

$$2 \cdot \frac{4}{3}s_1 - s_1^2 + \max\Big(0,\ (s_2 - s_1)\big(2 - \frac{s_2}{2} - s_1\big)\Big) \leq 2$$

with equation if $s_1 = 0$ and $s_2 = 2$. This contradicts with (2).

Case 2. $\mathrm{supp}(\mu) \neq [0,1]$. We only consider the case when there exist $0 < a < b < 1$ satisfying $\mu(a,b) = \mu(1-b,\ 1-a) = 0$, but $\mu(a), \mu(b) > 0$. The proofs of the other cases are essentially the same.

Compare the score of the two hiding strategies $H_1 = (a,\ 1-a)$ and $H_2 = \big(a,\ 1 - \frac{a+b}{2}\big)$ against S. Compare them when the same pairs of holes (x_1, x_2) are chosen. Lemma 2.10 implies that the first nut is found at the same time point and in the same hole in the two cases. If this is the first nut (at (x_1, a)), then Lemma 2.9 shows that the searcher digs either both or none of the two points $\big(x_2,\ 1 - \frac{a+b}{2}\big)$ and $(x_2,\ 1-a)$, therefore, H_1 and H_2 are equally good in this case. If the second nut was found for first, then

the other nut is in the same place (x_1, a) in the two cases, and Lemma 2.9 shows that S plays optimally against H_1 after finding this nut. This implies that S gets at most as much score against H_2 than against H_1.

The optimality of S and μ with the fact that $\mu(1-a) = \mu(a) > 0$ imply that S gets the score v against H_1. But S gets at least v against all hiding strategies. Therefore, S gets the score v against H_2, as well. This means that if S finds the second nut, then he plays optimally also against H_2, meaning that it no longer digs deeper than a.

We can use the same argument with $H_2' = \left(\frac{a+b}{2},\ 1-a\right)$. Therefore, if the hider chooses depths $\left(\frac{a+b}{2},\ 1-\frac{a+b}{2}\right)$, then S, after finding one nut, completely fails to dig at the right depth for the other nut, and hereby the searcher gets the score 0. This contradicts with the optimality of S. □

2.1 Solution for the double limit game

For first, it seemed that the extremal double limit game is easier to solve than the double limit game. The reason of it is that the strategy of the hider is a probability distribution on an interval in the extremal case, and on a two-dimensional domain in the non-extremal case. And therefore, the author expected the extremal game to be easier to solve than the original game. But the truth seems to be the opposite.

The extremal double limit game is not solved yet. If somebody tries to solve it, then the author suggests considering the searcher's function $f_{t-1}(x) = \chi(x < t) \cdot \left(1 - \frac{1}{t}\right)$ with probability more than $\frac{1}{2}$. The searcher's other pure strategy may start with $f_t(x) = \chi(x < t)g(x)$ for $t \in [0, \varepsilon]$ with a function $g(0) = 1$, $g'(0) \approx -0.1$. The author believes that $v_X(2,2) \approx 2.8$.

On the other hand, the double limit game has a surprisingly simple solution, as follows.

Theorem 2.13. *If $\lambda \geq k = j$, then $v(k,j,\lambda) = v(k,k,\lambda) = \frac{k!}{\lambda^k}$, and therefore, the value of the double limit game is $v(k,k) = k!$.*

The proof of Theorem 2.13 consists of Propositions 2.14 and 2.15, showing optimal strategies for the hider and the searcher.

Proposition 2.14. *In $LG(k,k,\lambda)$ with $\lambda \geq k$, if the hider chooses a uniform random point $(y_1, y_2, ..., y_k)$ from the simplex $y_i > 0$ $(\forall i \in \{1,2,...,k\})$, $\sum y_i \leq 1$, then the searcher wins with probability at most $\frac{k!}{\lambda^k}$. (Moreover, the searcher wins with exactly this probability provided that he always searches for nuts in places where it is possible to find one (e.g. never in depth > 1).)*

Proof. Consider the measure space T of k time points $0 \leq t_1 \leq t_2 \leq ... \leq t_k \leq 1$. For each strategy of the searcher, consider the measure space S of all allocations of the nuts for which the searcher would find all nuts. To each allocation in S, we can assign the vector of time points when the searcher finds the nuts. This is an injective mapping from S to T, and the inverse of it is measure-preserving. Therefore, the measure of S is at most the measure of T. The allocation of the nuts is a uniform random point $(x_1, y_1, x_2, y_2, ..., x_k, y_k)$ from the set $x_i \in [0, \lambda]$, $y_i > 0$, $\sum y_i \leq 1$, but this set is factored by the $k!$ permutations of the k indices. Therefore, the winning probability of the searcher is at most

$$\frac{\operatorname{Vol}\left((t_1,t_2,...,t_k) \;\middle|\; 0 \leq t_1 \leq t_2 \leq ... \leq t_k \leq 1\right)}{\frac{1}{k!} \cdot \lambda^k \cdot \operatorname{Vol}\left((y_1,y_2,...,y_k) \;\middle|\; (\forall i\colon y_i \geq 0), \sum y_i \leq 1\right)}$$

$$= \frac{k! \operatorname{Vol}\left((t_1,t_2,...,t_k) \;\middle|\; t_1, t_2 - t_1, t_3 - t_2, ..., t_k - t_{k-1} \geq 0; t_k \leq 1\right)}{\lambda^k \cdot \operatorname{Vol}\left((y_1,y_2,...,y_k) \;\middle|\; (\forall i\colon y_i \geq 0), \sum y_i \leq 1\right)} = \frac{k!}{\lambda^k}. \tag{3}$$

□

Proposition 2.15. *In $LG(k,k,\lambda)$ for $\lambda \geq k$, the searcher can win with probability at least $\frac{k!}{\lambda^k}$ by the following strategy.*

He digs parallelly in a unit interval, and if he finds a nut, then he goes to the next interval. Formally, if he found so far q nuts at the points in time $t_1, t_2, ..., t_q$, then with $t_0 = 0$, he chooses the function

$$f_t(x) = \sum_{i=1}^{q} \left(\chi(i-1 \leq x < i) \cdot (t_i - t_{i-1})\right) + \chi(q \leq x < q+1) \cdot (t - t_q).$$

Proof. If there is a group of nuts in each of the intervals $[0,1)$, $[1,2)$, ... $[k'-1,k')$, then the searcher finds all nuts. This has a probability $\frac{k'!}{\lambda^{k'}} \geq \frac{k!}{\lambda^k}$. □

Theorem 2.16. *If $k \leq 3$, then $v(k,k-1) = k!$.*

Proof. The same strategy of the searcher as in $DLG(k,k)$ provides a lower bound of $k \cdot (k-1)! = k!$.

If the hider chooses $(y_1, y_2, ..., y_k)$ uniformly randomly from the simplex $y_i \geq 0$, $\sum y_i = 1$, then the joint distribution of the $k-1$ variables, say $(y_1, y_2, ..., y_{k-1})$ is just a uniform random vector from the simplex $y_i \geq 0$, $\sum_{i=1}^{k-1} y_i \leq 1$. Therefore, this shows an upper bound of $k \cdot (k-1)! = k!$, as well. □

2.2 The discrete limit game

As we will see, the solution for the double limit game for $k = j$ is conjectured and partially proved to work if h is larger than a constant. Therefore, it will be useful to define a limit game when $n \to \infty$ and k, j and h are constant.

Definition 2.17. *The **discrete limit game** $DG(k,j,h)$ is defined as follows. The hider chooses k values $y_1, y_2, ...y_k \in [0,1]$ where $\sum y_i \leq 1$. Then for some subset $Q = \{q_1, q_2, ..., q_j\} \subset \{1,2,...,k\}$ with a vector of different positive integers $x_{q_1}, x_{q_2}, ...x_{q_j} \in \mathbb{Z}^+$, the nuts are placed at (x_{q_i}, y_{q_i}). The searcher observes nothing. Independently from this, the searcher defines a strategy of the limit game with $\lambda = \infty$. The score of the searcher is the number of the vectors $x_{q_1}, x_{q_2}, ...x_{q_j}$ for which he wins, summing up for all different Q. This is what the searcher aims to maximize and the hider aims to minimize, in expectation. The value of this game is denoted by $v^*(k,j,h)$.*

Theorem 2.18.

$$v^*(k,j,h) = \lim_{n\to\infty} v(k,j,n,h).$$

Sketch of proof. The proof will be similar to the proof for LG in Theorem 2.6 with the following step from Theorem 2.8.

In $G(k,j,n,h)$, by hiding the nuts into random holes with depths $\frac{1}{k}$, we can get that $v(k,j,n,h) = O(n^{-j})$. On the other hand, the searcher can get a score $\Omega(n^{-j+1})$ against placements that use the same hole for at least two nuts. E.g. $f_1(x) = \chi(x < 1)$ makes the job. Therefore, in any optimal hiding strategy, the probability that all nuts are placed in different holes should tend to 1.

Similar argument shows that if $n \to \infty$ and $\varepsilon \to 0$, then the probability that the hider chooses a depth less than ε should also tend to 0. This implies that forbidding the searcher to dig in more than $\frac{1}{\varepsilon}$ holes has a negligible effect on the value of the game.

Now we can convert the optimal hiding and searching strategy of DG to strategies of G with almost the same minimax and maximin scores with an error tending to 0. This works in the same way as in Theorem 2.6. □

Theorem 2.19. $v(k,j,n,h) \leq \binom{n+j-1}{j} \cdot v^*(k,j,h)$

Sketch of proof. First, we bound the winning probability by the expected number of j-element subsets of nuts that would have been found by the searcher if the other nuts had not been cached.

Consider a hiding strategy of DG, and choose the same hiding strategy in G in the following sense. The hider chooses a uniformly random k-element multiset of holes out of the $\binom{n+k-1}{k}$ possibilities. We put the nuts into these holes in different depths as follows. We will have k distances: the depth of the first nut in each non-empty hole, and the additional depths of the further nuts from the previous nuts. These depths will be a uniform random permutation of the random depths used by the optimal hiding strategy in DG.

Now any strategy of the searcher in DG can be transformed to a strategy in G by instead of digging a hole after finding a nut, the searcher digs in a new hole. This transformed strategy wins in the same number of cases. □

Conjecture 2.20. *If $k = j$ and for any h, if n is large enough, then the bound in Theorem 2.19 is sharp, and hereby the transformed hiding strategy in G is optimal.*

Question 2.21. *Is Conjecture 2.20 true for all $k = j > 2$?*

2.3 Solutions for the original problem for $k = j = 2$

In this section, unless we say the opposite, we **always assume that $k = j = 2$**, namely, the hider caches two nuts, and the searcher aims to find both of them. First, we present the following version of Conjecture 2.20 which will simplify further analysis.

Conjecture 2.22. *For any n and h, there always exists an optimal hiding strategy which is a probability distribution on the following basic strategies, denoted by pairs (y_1, y_2). With such a basic strategy, the searcher chooses two holes, maybe the same hole twice, uniformly randomly out of the $\binom{n+1}{2}$ choices. If he chooses two different holes $x_1 \neq x_2$, then he caches the two nuts to (x_1, y_1) and (x_2, y_2), or (x_1, y_2) and (x_2, y_1), with the same probabilities. If he chooses the same hole x, then he caches the nuts to (x, y_1) and $(x, 1)$, or (x, y_2) and $(x, 1)$, randomly.*[2]

This conjecture does not seem to be very difficult to prove. It would also be interesting whether we can say anything similar for other values of k and j.

In the light of this conjecture, we can use the solution of the double limit game (Theorem 2.13) for the original game as follows.

Theorem 2.23. *If the hider uses the strategy (y_1, y_2) for a uniform random pair satisfying $y_1 \geq 0$, $y_2 \geq 0$, $y_1 + y_2 \leq 1$, then the searcher wins with probability at most $\frac{2h^2}{n(n+1)}$. If $h \in \mathbb{Z}^+$ and $h \leq \frac{n+1}{2}$, then the bound is sharp, namely, the value of the game is $\frac{2h^2}{n(n+1)}$.*

The proof will be very similar to the proof of Theorem 2.13. It will follow from Theorem 2.36 and Theorem 2.37, about the strategies of the hider and the searcher.

Conjecture 2.24. *The bound $\frac{2h^2}{n(n+1)}$ in Theorem 2.23 is sharp if $\frac{h^2}{\lfloor h \rfloor} \leq \frac{n+1}{2}$ and either $h \geq 3$ or $h = 3 - \frac{1}{q}$ for any $q \in \mathbb{Z}^+ \setminus \{3\}$.*

Theorem 2.25. *If $\frac{n+1}{2} \leq h$, then the value of the game is $\frac{\lfloor h \rfloor}{n}$. This is always an upper bound for the value of the game, because of the hiding strategy of putting both nuts at the same random hole, at depth 1.*

Proof. Hiding both nuts at the same hole in depth 1 is provides hiding probability at most $\frac{\lfloor h \rfloor}{n}$.

Consider now the following strategy of the searcher. He chooses $\lfloor h \rfloor$ holes at random, and starts digging in them parallelly, until a nut is found, at hole x in depth y. Then he continues digging x until depth 1. Then if $y \leq \frac{1}{2}$, then he digs the other $\lfloor h \rfloor - 1$ chosen holes until depth $1 - y$, and the remaining $n - \lfloor h \rfloor$ holes until depth $\min(y, 1 - y)$.

If the nut with the higher depth (if the depths are the same, then either nut) is in one of the $\lfloor h \rfloor$ chosen holes, then the searcher finds both nuts. This has a probability at least $\frac{\lfloor h \rfloor}{n}$.

This strategy uses a total digging amount of

$$1 + \big(\lfloor h \rfloor - 1\big) \cdot \max(y, 1-y) + \big(n - \lfloor h \rfloor\big) \cdot \min(y, 1-y)$$

$$= 1 + (n-1) \cdot \min(y, 1-y) + \big(\lfloor h \rfloor - 1\big) \cdot \big(\max(y, 1-y) - \min(y, 1-y)\big)$$

$$\leq 1 + (2h-2) \cdot \min(y, 1-y) + (h-1) \cdot \big(\max(y, 1-y) - \min(y, 1-y)\big)$$

$$= 1 + (h-1) \cdot \big(\max(y, 1-y) + \min(y, 1-y)\big) = 1 + (h-1) = h. \qquad \square$$

[2] The strategy $(1, 0)$ can be replaced by the strategy of caching both nuts in the same random hole in depth 1.

Conjecture 2.26. *If* $\frac{n+1}{2} \leq \frac{h^2}{\lfloor h \rfloor}$*, then the value of the game is* $\frac{\lfloor h \rfloor}{n}$*.*

To challenge Conjectures 2.24 and 2.26, or to try to prove them, the author suggests considering the following question.

Question 2.27. *For* $n = 6$*,* $h = \sqrt{10.5} \approx 3.24$*, is it true that the searcher can win with probability* $\frac{1}{2}$*?*

If $h < 3$, then the following discrete version of the searcher's double limit game solution can provide a better upper bound.

Theorem 2.28. *If* $h < \frac{a}{b}$ *for some* $a, b \in \mathbb{Z}^+$*, then with the following hiding strategy, the searcher always wins with probability at most* $\frac{2(a-1)(a-2)}{b(b-1)\cdot n(n+1)}$*.*

The hider chooses $y_1, y_2 \in \left\{\frac{1}{b}, \frac{2}{b}, \frac{3}{b}, ..., \frac{b-1}{b}\right\}$*,* $y_1 + y_2 \leq 1$ *uniformly at random from the* $\binom{b}{2}$ *possible choices, and chooses the hiding straregy* (y_1, y_2)*.*

Proof. The searcher can dig at most $a-1$ possible hiding points (depths $\frac{1}{b}, \frac{2}{b}, ..., 1$). Given the strategy of searcher, if he finds the two nuts at the ith and jth searched possible hiding points, then it determines the two positions of the nuts. There are $\binom{a-1}{2}$ different pairs of integers $1 \leq i < j \leq a-1$, and there are $\binom{b}{2} \cdot \binom{n+1}{2}$ possible pairs of positions, so the searcher cannot win with higher probability than $\frac{\binom{a-1}{2}}{\binom{b}{2}\binom{n+1}{2}} = \frac{2(a-1)(a-2)}{b(b-1)\cdot n(n+1)}$. □

Conjecture 2.29. *If* $h \in \left(\frac{5}{2}, \frac{8}{3}\right) \cup \left[\frac{19}{7}, 2\right) \setminus \left\{3 - \frac{1}{q} : q \in \mathbb{Z}^+\right\}$*, then the best upper bound provided by Theorem 2.28 is sharp.*

Now we have a conjecture of the solution for $h \in \left[\frac{5}{2}, n\right] \setminus \left[\frac{8}{3}, \frac{19}{7}\right)$.

For $h \in \left[0, \frac{9}{5}\right) \cup \left[2, \frac{7}{3}\right)$, the values of the games are the very same as for $n = 4$, written in the form $\frac{\alpha(h)}{n(n+1)}$. The proofs are also essentially the same.

Theorem 2.30. *For* $h \in \left[2 - \frac{1}{q-1},\ 2 - \frac{1}{q}\right)$*,* $q \in \{5, 6, 7, 8, 9\}$ *and* $n \leq q-1$*, and for* $h \in \left[\frac{9}{5}, 2\right)$ *and* $n \leq 5$*, the values of the games are* $\frac{9}{2}, 5, \frac{26}{5}, \frac{28}{5}, \frac{17}{3}, 6$*, respectively, divided by* $n(n+1)$*.*

An optimal hiding strategy in the first 5 cases are $\left(\frac{1}{q}, \frac{q-1}{q}\right)$ *with probability* $\frac{1}{2}, \frac{1}{2}, \frac{2}{5}, \frac{2}{5}, \frac{1}{3}$*, respectively, and* $\left(\frac{q-1}{2q}, \frac{q+1}{2q}\right)$ *otherwise. In the last case, it is* $\left(\frac{1}{4}, \frac{3}{4}\right)$ *with probability* $\frac{2}{3}$ *and* $\left(\frac{1}{2}, \frac{1}{2}\right)$ *with probability* $\frac{1}{3}$*, or in other words, it is a uniform random extremal strategy with depths multiples of* $\frac{1}{4}$*. An optimal strategy of the searcher is the mixture of the followings, until finding the first nut (the continuation is obvious, see Lemma 2.9). He caches a random hole until depth* $h-1$*, then another one until depth* $\frac{3-h}{2}$*, then continues the first hole until depth 1. Or he just caches a random hole until depth* 1*. The former strategy is used with probabilities* $\frac{1}{4}, \frac{2}{4}, \frac{3}{5}, \frac{4}{5}, \frac{5}{6}, 1$*, respectively.*

The proof is a simple but long case analysis which we omit from this paper.

For $h \in \left[\frac{7}{3}, \frac{5}{2}\right)$, we expect a similar but more difficult structure of the solutions as for $h \in \left[\frac{9}{5}, 2\right)$. What we know is the following.

Lemma 2.31. *If* $h < \frac{5}{2}$*, then the value of the game is at most* $\frac{11}{n(n+1)}$*. This can be achieved by the strategy* $\left(\frac{1}{4}, \frac{3}{4}\right)$ *with probability* $\frac{1}{2}$ *and* $\left(\frac{1}{2}, \frac{1}{2}\right)$ *with probability* $\frac{1}{2}$*.*

Conjecture 2.32. *The bound in Lemma 2.31 is optimal for* $h \in \left[\frac{17}{7}, \frac{5}{2}\right)$*.*

Note 2.33. *If one wants to solve* $h \in \left[\frac{7}{3}, \frac{17}{7}\right)$*, then we suggest considering mixtures of extremal hiding strategies* $\left(\frac{10-4h}{3}, \frac{4h-7}{3}\right)$*,* $\left(\frac{h-1}{5}, \frac{6-h}{5}\right)$*,* $\left(\frac{16-6h}{5}, \frac{6h-11}{5}\right)$*,* $\left(\frac{h-1}{3}, \frac{4-h}{3}\right)$*.*

Theorem 2.34. *If* $h < h^* = \frac{67}{25}$ *or* $\frac{51}{19}$ *or* $\frac{19}{7}$*, and* $n \leq 11$*, then the searcher can win with probability at most* $\frac{14\frac{2}{53}}{n(n+1)}$*,* $\frac{14\frac{2}{27}}{n(n+1)}$*,* $\frac{14\frac{2}{11}}{n(n+1)}$*, respectively, if the hider uses the following mixture of hiding strategies.*

- $\left(\frac{3-h^*}{2}, \frac{h^*-1}{2}\right)$ *with probability* $\frac{12}{53}$*,* $\frac{20}{81}$*,* $\frac{4}{33}$*, respectively;*

- $\left(\frac{h^*-1}{6}, \frac{7-h^*}{6}\right)$ *with probability* $\frac{4}{53}$, $\frac{4}{81}$, $\frac{4}{33}$, *respectively;*
- $\left(\frac{h^*-1}{4}, \frac{5-h^*}{4}\right)$ *with probability* $\frac{36}{53}$, $\frac{56}{81}$, $\frac{8}{11}$, *respectively;*
- $\left(\frac{h^*-1}{6}, \frac{h^*-1}{6}\right)$ *with probability* $\frac{1}{53}$, $\frac{1}{81}$, $\frac{1}{33}$, *respectively.*

In particular, in the third case, the four depths are $\left(\frac{1}{7}, \frac{6}{7}\right)$, $\left(\frac{2}{7}, \frac{5}{7}\right)$, $\left(\frac{3}{7}, \frac{4}{7}\right)$ *and* $\left(\frac{2}{7}, \frac{2}{7}\right)$.

The proof again is a simple but long case analysis, which we omit from this paper.

Conjecture 2.35. *If* $h \in \left[\frac{8}{3}, \frac{19}{7}\right)$, *then the best bound in Theorem 2.34 is sharp.*

The table summarizes our results and conjectures for $k = j = 2$.

h	$v(2,2,n,h)$	validity	status	notes
$[0,1)$	0	every n	proved	proved in earlier papers for $n=4$, the same proof works for $n \geq 4$
$\left[1, \frac{3}{2}\right)$	$\frac{2}{n(n+1)}$	$n \geq 2$		
$\left[\frac{3}{2}, \frac{5}{3}\right)$	$\frac{3}{n(n+1)}$			
$\left[\frac{5}{3}, \frac{7}{4}\right)$	$\frac{4}{n(n+1)}$	$n \geq 3$		
$\left[\frac{7}{4}, \frac{9}{5}\right)$	$\frac{4.5}{n(n+1)}$	$n \geq 4$		
$\left[\frac{9}{5}, \frac{11}{6}\right)$	$\frac{5}{n(n+1)}$	$n \geq 5$		proved in Theorem 2.30
$\left[\frac{11}{6}, \frac{13}{7}\right)$	$\frac{5.2}{n(n+1)}$	$n \geq 6$		
$\left[\frac{13}{7}, \frac{15}{8}\right)$	$\frac{5.6}{n(n+1)}$	$n \geq 7$		
$\left[\frac{15}{8}, \frac{17}{9}\right)$	$\frac{5\frac{2}{3}}{n(n+1)}$	$n \geq 8$		
$\left[\frac{17}{9}, 2\right)$	$\frac{6}{n(n+1)}$	$n \geq 5$		
$\left[2, \frac{11}{5}\right)$	$\frac{8}{n(n+1)}$	$n \geq 4$		proved in earlier papers for $n=4$, the same proof works for $n \geq 4$
$\left[\frac{11}{5}, \frac{7}{3}\right)$	$\frac{9}{n(n+1)}$			
$\left[\frac{7}{3}, \frac{17}{7}\right)$	?	$n \geq 5$	open	see Note 2.33
$\left[\frac{17}{7}, \frac{5}{2}\right)$	$\frac{11}{n(n+1)}$		upper bound proved	see Lemma 2.31 and Conjecture 2.32
$\frac{5}{2}$	$\frac{12.5}{n(n+1)}$	$n \geq 6$		see Theorem 2.23 and Conj. 2.24
$\left(\frac{5}{2}, \frac{8}{3}\right)$	$\inf_{\frac{a}{b}>h} \frac{2(a-1)(a-2)}{b(b-1)\cdot n(n+1)}$			see Theorem 2.28
$\left[\frac{8}{3}, \frac{67}{25}\right)$	$\frac{14\frac{2}{53}}{n(n+1)}$	$n \geq 11$		see Theorem 2.34 and Conj. 2.35
$\left[\frac{67}{25}, \frac{51}{19}\right)$	$\frac{14\frac{2}{27}}{n(n+1)}$			
$\left[\frac{51}{19}, \frac{19}{7}\right)$	$\frac{14\frac{2}{11}}{n(n+1)}$			
$\left[\frac{19}{7}, 3\right)$	$\inf_{\frac{a}{b}>h} \frac{2(a-1)(a-2)}{b(b-1)\cdot n(n+1)}$	$n \geq 8$		see Theorem 2.28 (and Conj. 2.24)
$\left\{3, 4, ..., \left\lfloor \frac{n+1}{2} \right\rfloor\right\}$	$\frac{2h^2}{n(n+1)}$	every n	proved	proved in Theorem 2.23
$\left(3, \frac{n}{2} + O(1)\right]$	$\frac{2h^2}{n(n+1)}$	$n \geq 6$	upper bound proved	see Conjecture 2.24
$\frac{n+1}{2} \leq \frac{h^2}{\lfloor h \rfloor}$	$\frac{\lfloor h \rfloor}{n}$	every n		see Conjecture 2.26
$\left[\frac{n+1}{2}, n\right]$			proved	proved in Theorem 2.25

2.4 Extensions for $k = j > 2$

Theorem 2.36. *If* $k \geq 2$, *then*

$$v(k,k,n,h) \leq \frac{h^k}{\binom{n+k-1}{k}}. \tag{4}$$

Proof. We convert the proof for $v(k,k)$ (Theorem 2.13) to a proof of this problem as follows. The hider chooses how many nuts to put to each hole, choosing one of the $\binom{n+k-1}{k}$ possibilities uniformly at random. Now we consider the distance of each nut from the closest nut above it, or if there is no nut above it, then the depth of the nut (the distance from the top). We choose these k depths $y'_1, y'_2, ...y'_k$ uniformly at random from the simplex $y'_i \geq 0$ $(\forall i \in \{1,2,...,k'\})$, $\sum y'_i \leq 1$. From here, we can continue with the proof of Proposition 2.14 with exchanging λ^k to $\binom{n+k-1}{k}$. □

Now we show that (4) is sharp if $k = 2$ and $h \in \mathbb{N}$.

Theorem 2.37. *If* $k \in \mathbb{N}$

$$v(2,2,n,h) \geq \frac{h^k}{\binom{n+k-1}{k}}.$$

Proof. Consider the following strategy of the searcher. He chooses h holes at random, and he is digging them parallelly until a nut is found (but until at most depth 1, when the game ends). If a nut is found at a depth y, then he chooses h new holes, as follows. With probability $\frac{2h}{n+1}$, he chooses the hole in which the nut was found, and the remaining $h-1$ or h holes are randomly chosen from the other $n-h$ holes. Then he digs these holes in depth $1-y$ (more).

Assume first that the hider caches the two nuts in two different holes. If exactly one of them are in one of the h holes the searcher started with, and the other one is in the next h holes, then the searcher finds both nuts. Therefore, the searcher finds both nuts in expectedly $h \cdot \left(h - \frac{2h}{n+1}\right)$ number of cases out of the $\binom{n}{2}$ pairs, which happens with probability

$$\frac{h \cdot \left(h - \frac{2h}{n+1}\right)}{\binom{n}{2}} = \frac{2h \cdot \frac{(n+1)h-2h}{n+1}}{n(n-1)} = \frac{2h \cdot \frac{(n-1)h}{n+1}}{n(n-1)} = \frac{2h^2}{n(n+1)}.$$

Assume now that the hider caches the two nuts in the same hole. If this nut is in the first h holes chosen by the searcher, and the searcher chooses to continue digging in this hole, then he finds both nuts. This has probability

$$\frac{h}{n} \cdot \frac{2h}{n+1} = \frac{2h^2}{n(n+1)}.$$

To sum up, this strategy of the searcher finds both nuts with probability at least $\frac{2h^2}{n(n+1)}$, against any strategy of the hider. □

Conjecture 2.38. *If* $k \geq 2$ *and* $\frac{k+1}{k-1} \leq h \leq \frac{n}{k}$, *then* $v(k,k,n,h) = \frac{h^k}{\binom{n+k-1}{k}}$.

Most probably, this conjecture can be proved for integral h. So this weaker, seemingly easier conjecture is the following.

Conjecture 2.39. *If* $k \geq 2$ *and* $h \in \mathbb{Z}^+$ *and* $h \leq \frac{n}{k}$, *then* $v(k,k,n,h) = \frac{h^k}{\binom{n+k-1}{k}}$.

The author believes that the same proof works here as for $k = 2$, in Theorem 2.37. Except that when the searcher finds a nut and chooses h new holes, then he might choose again any holes which had a nut at its current bottom. We should find the right probabilities for each of these choices so as for each distribution of the number of nuts in the different holes, the searcher finds them with the same probability.[3]

Note 2.40 (Final note for the section)**.** *There are many other potentially useful limit problems not yet considered, when* $k \to \infty$. *E.g. if* $j_i \to \infty$ *and* $\frac{j_i}{k_i}$ *is convergent. These limit games might also be very useful, and these may also look differently from the original game.*

[3]Update: Dömötör Pálvölgyi recently proved this conjecture, it will be published shortly.

3 Manickam–Miklós–Singhi Conjecture

As a much simpler application of the limit theory techniques, we show an easy way to generalize a well-known conjecture.

Problem 3.1 (MMS-Problem)**.** *For a fixed $n, k \in \mathbb{N}$, find a sequence $a_i \in \mathbb{R}$, $a_1+a_2+...+a_n = 0$ such that if $\{i_1, i_2, ..., i_k\} \subset \{1, 2, ..., n\}$ is a uniform random subset with cardinality k, then $\Pr\left(a_{i_1}+a_{i_2}+...+a_{i_k} > 0\right)$ is the largest possible.*

Denote this maximum probability by $M(n, k)$. Two sequences are equivalent if after applying a permutation on one sequence, the two events that the sum is positive are the same.

Conjecture 3.2 (Manickam–Miklós–Singhi)**.** *If $4k \leq n$, then $M(n, k) = \frac{n-k}{n}$. The only optimal solution up to equivalence is $1 - n, 1, 1, 1, ..., 1$.*

The Manickam–Miklós–Singhi Conjecture was introduced in 1987 in [10], and it has recently received a lot of attention, especially because of its connection to the Erdős matching conjecture [5]. In 2013, Huang and Sudakov [7] proved it for $33k^2 < n$. In 2014, Chowdhury, Sarkis and Shahriari [3] proved for $8k^2 < n$. Then in 2015, Pokrovskiy [11] proved for $k < \varepsilon \cdot n$, but this improves the previous results only for $k > 10^{45}$.

We consider a limit problem when $n \to \infty$, and $\frac{k}{n}$ converges. Finding the right limit problem is not an obvious task. One of the problems is that the solution $1 - n, 1, 1, 1, ..., 1$ does not have a limit when $n \to \infty$. We resolve this problem by dropping the condition $a_1 + a_2 + ... + a_n = 0$, but we want to maximize $\Pr\left(a_{i_1} + a_{i_2} + ... + a_{i_k} > \frac{k}{n}\sum a_i\right)$. This is clearly equivalent to the original problem. Now $-1, 0, 0, ..., 0$ is an equivalent form of the conjectured optimal solution, and we can say that the infinite sequence $-1, 0, 0, ...$ is a limit of it. We needed a few more observations to define the following limit problem.

Problem 3.3. *For a fix $p \in (0, 1)$, we are looking for a countable sequence $a_1, a_2, ...$ of real numbers with $\sum a_i^2 < \infty$ and a real number d which maximizes $\Pr\left(\sum a_i(x_i - p) + dx_0 > 0\right)$, where $x_1, x_2, ...$ are indicator variables with probability p, and x_0 is a variable with standard normal distribution, and $x_0, x_1, x_2, ...$ are independent.*

Denote this supremum probability by $M(p)$. (Which is a maximum, but we do not prove it here.) We call this problem a limit problem of the Manickam–Miklós–Singhi Conjecure, because the following theorem holds.

Theorem 3.4. *For any sequence (n_i, k_i), $n_i \to \infty$ and $\frac{k_i}{n_i} \to p \in (0, 1)$,*

$$\liminf_{\delta \to 0} M(p + \delta) \leq \liminf M(n_i, k_i) \leq \limsup M(n_i, k_i) \leq \limsup_{\delta \to 0} M(p + \delta).$$

Sketch of proof. We can naturally convert a solution of the finite problem to a solution of the limit problem and vice versa. We only need to prove that the conversion error tends to 0 when $n \to \infty$.

The conversion of a solution of the finite problem (with large n and k) to a solution of the limit game is essentially the following. We normalize the finite sequence by making its median 0. The rest of the terms is 0 and $d = 0$.

The conversion of a solution of the limit problem to a solution of a large finite problem is a bit more tricky. We keep a finite number of terms a_i with the largest absolute values. Half of the rest of the terms will be ε and the other half will be $-\varepsilon$, where the value ε is chosen so as to keep the total variance the same as it was in the limit problem. Using the following version of the Central Limit Theorem, we can deduce that this conversion error also tends to 0.

Lemma 3.5. *For every $p \in (0, 1)$ and $\varepsilon > 0$, there exists $\delta > 0$ such that for any sequence $-\delta < a_1, a_2, ..., a_n < \delta$ and $\left|\frac{k}{n} - p\right| < \delta$ and $t \in \mathbb{R}$, the following holds. If $\{i_1, i_2, ..., i_k\} \subset \{1, 2, ..., n\}$ is a*

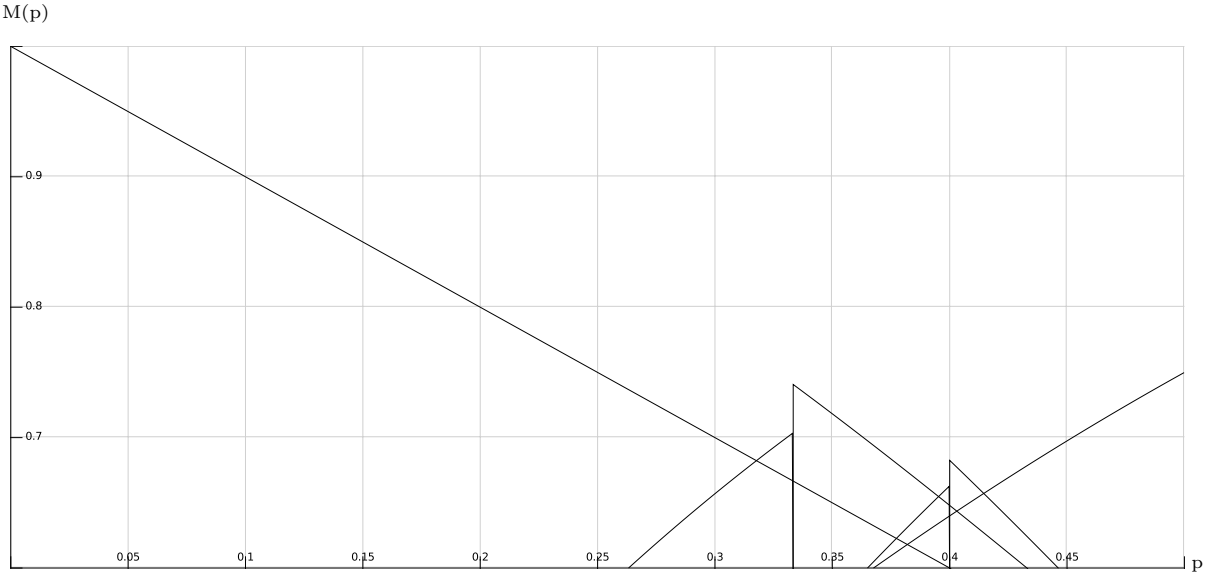


Figure 1. $M(p)$ is the maximum of the functions in the figure. If p is between 0 and 0.317... (or exactly $1/3$), then -1 is the best (and the other coefficients are 0), between 0.317... and $1/3$ the sequence $1, 1, 1$ is the best, between $1/3$ and 0.395... (and at 0.4) the sequence $-1, -1, -1$, between 0.395... and 0.4 the sequence $1, 1, 1, 1, 1$, between 0.4 and 0.414... the sequence $-1, -1, -1, -1, -1$, and between 0.414... and 0.5 the sequence $1, 1$.

uniform random subset with cardinality k, then

$$\Phi\big(\sigma(t-\varepsilon)\big) - \varepsilon < \Pr\Big(a_{i_1} + a_{i_2} + ... + a_{i_k} < \frac{k}{n}\sum_{i=1}^{n} a_i + t\Big) < \Phi\big(\sigma(t+\varepsilon)\big) + \varepsilon,$$

where $\sigma^2 = \operatorname{Var}\big(a_{i_1} + a_{i_2} + ... + a_{i_k}\big)$ and Φ is the distribution function of the standard normal distribution. □

We analyzed the limit problem for all values of p, the results are summarized in Figure 1. This leads to a new conjecture as follows.

Conjecture 3.6. *The optimal solution of the limit game has the form $a_1 = a_2 = ... = a_q$ where $q \in \{1, 2, 3, 5\}$, and all other coefficients are 0. This is the only optimal solution up to equivalence.*

Now we are ready to form the corresponding conjecture for the original problem.

Conjecture 3.7. *The optimal solution of the original finite game has the form $a_1 = a_2 = ... = a_q$ where $q \in \{1, 2, 3, 5\}$, and $a_{q+1} = a_{q+2} = ... = a_n = -\frac{q}{n-q}a_1$. This is the only optimal solution up to equivalence.*

Now we show a possible way to prove the conjecture by analysing a very large but finite number of cases (most probably using a computer). We say that a feasible normalized solution, or in short, a *solution* for the limit problem (Problem 3.3) is a sequence and a number $\big((a_i)_{i\in\mathbb{N}}, d\big)$, where

$$\operatorname{Var}\Big(\sum a_i(x_i - p) + dx_0\Big) = p(1-p)\sum_{i\in\mathbb{N}} a_i^2 + d^2 = 1.$$

Lemma 3.8. *We define the distance between two strategies $\big((a_i), d_\alpha\big)$ and $\big((b_i), d_\beta\big)$ by*

$$\inf_{\pi_1,\pi_2,k}\Big(\sum_{i=1}^{k}\big|a_{\pi_i} - b_{\pi_i}\big| + \sup_{i>k}\big|a_{\pi_i}\big| + \sup_{i>k}\big|b_{\pi_i}\big|\Big),$$

where π_1 and π_2 are permutations on $\mathbb{Z}^+$. The topological space of the strategies induced by this distance function is compact.

Lemma 3.9. *If for a solution $s = \big((a_i), d\big)$, $\Pr\big(\sum a_i(x_i - p) + dx_0 > -\varepsilon\big) < v$, then there exists a neighborhood of s in which for every solution $\big((a'_i), d'\big)$, we have $\Pr\big(\sum a'_i(x_i - p) + d'x_0 > 0\big) < v$.*

With these two lemmas, we can hope that we can cover the solution space with a finite number of regions (open sets) so that we can show the conjectured inequality in each of these regions. Then we could modify these proofs so as to make it valid for the original discrete problem, as well. For this, we would also need a version of Lemma 3.5 which gives an explicit $\delta > 0$ for each $\varepsilon > 0$.

4 Kikuta–Ruckle Conjecture

We can use the same technique for the following generalization of the MMS-Problem, defined by Kikuta and Ruckle. [1, 8]

Problem 4.1 (KR-Problem)**.** *$n, k \in \mathbb{N}$, and $d \in (0, 1)$ are given. We want to find nonnegative real numbers $a_1, a_2, ..., a_n \geq 0$ with $a_1 + a_2 + ... + a_n = 1$ which maximizes $\Pr\big(a_{i_1} + a_{i_2} + ... + a_{i_k} > d\big)$, where $\{i_1, i_2, ..., i_k\} \subset \{1, 2, ..., n\}$ is a uniform random k-element subset.*

Denote this supremum[4] probability by $K(k, n, d)$. Notice that if $d < \frac{k}{n}$, then $a_i = \frac{1}{n}$ provides $K(k, n, d) = 1$ and if $d = \frac{k}{n}$, then we get back the MMS-Problem.

Furthermore, we can get the Kikuta–Ruckle problem from Alpern's Caching Game by the following modification and by taking the limit $k \to \infty$ and $\frac{k}{n} \to d$. This modification is that we replace the overall hiding time limit to the restriction that the searcher cannot use a depth more than 1 (or other than 1), and we consider the limit $k \to \infty$ and $\frac{j}{k} \to d$.

Conjecture 4.2 (Kikuta–Ruckle)**.** *For all $n, k \in \mathbb{N}$, and $d \in (0, 1)$, there is an optimal solution for the KR-Problem of the form $a_1 = a_2 = ... = a_s = \frac{1}{s}$ and $a_{s+1} = a_{s+2} = ... = a_n = 0$ for some $s \in \{1, 2, ..., n\}$.*

The conjecture says nothing about the optimal value s. The authors as well as other researchers on the topic found a very chaotic behaviour of this value. However, it would be useful to know the value if we want to prove the conjecture. Searching for the optimal values for small constant values n, k did not really help, we will shortly see the reason of it. Instead, we will consider what happens if $n \to \infty$ and hereby we form a conjecture about the value of s.

The KR-Problem has one more parameter than the MMS-Problem, therefore, we have a larger freedom about defining limit problems of it with $n_i \to \infty$. The simplest limit problem is when $\frac{k_i}{n_i} \to p \in (0, 1)$ and $d_i = d$. The conjectured value of s is described in Figure 2.

However, it misses the most important cases: when d is just above $\frac{k}{n}$. One of the most important limits of the case is $n \to \infty$, $\frac{k}{n} \to p \in (0, 1)$, $d = \frac{k}{n}$, which is just Problem 3.3. But if $d > \frac{k}{n-1}$, then $s = n - 1$, or the corresponding solution $-1, 0, 0, ...$ for the MMS-problem is no longer good. Moreover, if $d > \frac{k}{n-c}$ for any constant c, then $s \leq n - c$ provides probability 0. Therefore, if $n \to \infty$, $\frac{k}{n} \to p \in (0, 1)$, $d \to p$ and $n - \frac{k}{d} \to \infty$, then this leads to the following limit problem.

Problem 4.3. *For a fix $p \in (0, 1)$, we are looking for a countable sequence $0 \leq a_1, a_2, ...$ of* nonnegative *real numbers with $\sum a_i^2 < \infty$ and a real number σ which maximizes*

$$\Pr\big(\sum a_i(x_i - p) + \sigma x_0 > 0\big), \tag{5}$$

where $x_1, x_2, ...$ are indicator variables with probability p, and x_0 is a variable with standard normal distribution, and $x_0, x_1, x_2, ...$ are independent.

[4] We believe that this is a maximum. If we use "$\geq d$" rather than "$> d$", than due to the compactness of the space of solutions, we can even prove it.

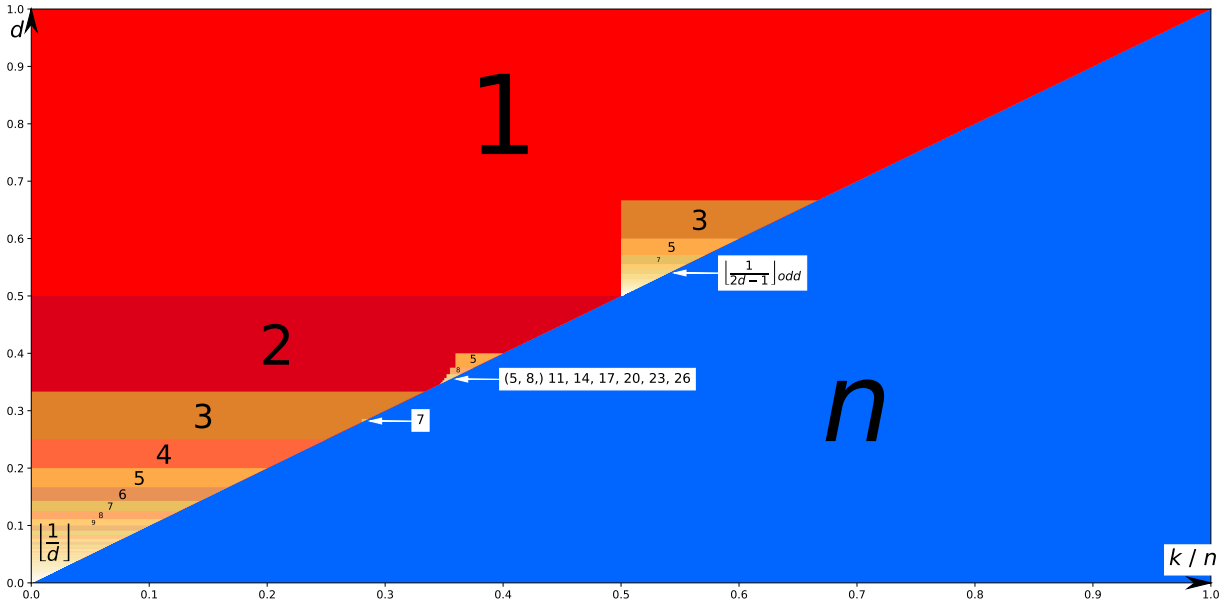


Figure 2

Denote this supremum probability by $K(p)$.

Again, the reason why we consider it a true limit problem is the following theorem.

Theorem 4.4. *For any sequence (n_i, k_i, d_i), $n_i \to \infty$ and $\frac{k_i}{n_i} \to p \in (0,1)$, $d_i \to p$ and $n_i - \frac{k_i}{d_i} \to \infty$,*

$$\liminf_{\delta\to 0} K(p+\delta) \le \liminf K(n_i,k_i,d_i) \le \limsup K(n_i,k_i,d_i) \le \limsup_{\delta\to 0} K(p+\delta).$$

Sketch of proof. The proof is the same as for Theorem 3.4 with the additional observation that the median is at most $\frac{2}{n}$, and changing a few terms a_i by $O(\frac{1}{n})$ has a negligible effect. □

Another limit problem is the following. If $0 < \frac{k}{n} < d \to 0$, with different values of $\lim dn - k = \lambda$, then it has the same limit as Problem 4.3 except that (5) is replaced to the following.

$$\Pr\big(\sum a_i(x_i - p) + \sigma x_0 > -\lambda \min_i a_i\big)$$

After analysing all limits, we can make a conjecture about how s depends on n and d.

Conjecture 4.5. *For all $n, k \in \mathbb{N}$, and $d \in (0,1)$, there is an optimal solution for the KR-Problem of the form $a_1 = a_2 = \ldots = a_s = \frac{1}{s}$ and $a_{s+1} = a_{s+2} = \ldots = a_n = 0$, where*[5]

$$s \in \Big\{\Big\lfloor\frac{1}{d}\Big\rfloor, \Big\lfloor\frac{1}{2d-1}\Big\rfloor_{\text{odd}}, 5, 7, 8, 11, 14, 17, 20, 23, 26, \Big\lfloor\frac{k}{d}\Big\rfloor, \Big\lfloor\frac{2k-n}{2d-1}\Big\rfloor_{\equiv n \pmod 2},$$
$$n-10, n-7, n-6, n-5, n-4, n-3, n\Big\}.$$

The author found this technique very useful for seeking for counterexamples, as well. Now he strongly believes that the conjecture is true, but it is rather "accidentally true" and he doubts that there exists a simple proof. He found that the best candidates for counterexamples use the terms $\frac{2}{s}$, $\frac{1}{s}$ and 0 for some s. Showing that there is no counterexample of this form is already a very difficult task, we need completely different arguments for the different cases.

[5] $\lfloor x \rfloor = \max\{y \in \mathbb{Z} : y < x\}$ and $\lfloor x \rfloor_{\text{odd}} = \max\{y \equiv 1 \pmod 2, y < x\}$ and $\lfloor x \rfloor_{\equiv n \pmod 2} = \max\{y \equiv n \pmod 2, y < x\}$. If we define the KR-Problem with "$\le$" instead of "$<$", then here we have $y \le x$.

References

[1] Steve Alpern, Robbert Fokkink, G Leszek, Roy Lindelauf, VS Subrahmanian, et al. *Search theory: a game theoretic perspective*. Springer Science & Business Media, 2013.

[2] Steve Alpern, Robbert Fokkink, Thomas Lidbetter, and Nicola S Clayton. A search game model of the scatter hoarder's problem. *Journal of The Royal Society Interface*, 9(70):869–879, 2012.

[3] Ameera Chowdhury, Ghassan Sarkis, and Shahriar Shahriari. The Manickam–Miklós–Singhi conjectures for sets and vector spaces. *Journal of Combinatorial Theory, Series A*, 128:84–103, 2014.

[4] Endre Csóka and Thomas Lidbetter. The solution to an open problem for a caching game. *Naval Research Logistics (NRL)*, 2016.

[5] Pál Erdős. A problem on independent r-tuples. In *ARTICLE IN PRESS B. Bollobás et al./Journal of Combinatorial Theory, Series A*. Citeseer, 1965.

[6] Harry Furstenberg. Ergodic behavior of diagonal measures and a theorem of szemerédi on arithmetic progressions. *Journal d'Analyse Mathématique*, 31(1):204–256, 1977.

[7] Hao Huang and Benny Sudakov. The minimum number of nonnegative edges in hypergraphs. *The Electronic Journal of Combinatorics*, 21(3):P3–7, 2014.

[8] Kensaku Kikuta and William H Ruckle. Continuous accumulation games on discrete locations. *Naval Research Logistics (NRL)*, 49(1):60–77, 2002.

[9] László Lovász and Balázs Szegedy. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, 96(6):933–957, 2006.

[10] N Manickam and D Miklós. On the number of non-negative partial sums of a non-negative sum. In *Colloq. Math. Soc. János Bolyai*, volume 52, pages 385–392, 1987.

[11] Alexey Pokrovskiy. A linear bound on the Manickam–Miklós–Singhi conjecture. *Journal of Combinatorial Theory, Series A*, 133:280–306, 2015.

Linear cycle-free hypergraphs, covers by linear cycles

Beka Ergemlidze

Department of Mathematics
Central European University, Budapest
beka.ergemlidze@gmail.com

Ervin Győri

Rényi Institute, Hungarian Academy of Sciences
and
Department of Mathematics
Central European University, Budapest
gyori.ervin@renyi.mta.hu

Abhishek Methuku

Department of Mathematics
Central European University, Budapest
abhishekmethuku@gmail.com

Abstract: In this paper we continue the work of Gyárfás, Győri and Simonovits [3], who proved that if a 3-uniform hypergraph H with n vertices has no linear cycles, then its independence number $\alpha \geq \frac{2n}{5}$. The hypergraph consisting of vertex disjoint copies of complete hypergraphs K_5^3 shows that equality can hold. They asked whether α can be improved if we exclude K_5^3 as a subhypergraph and whether such a hypergraph is 2-colorable.

In this paper, we answer these questions affirmatively. Namely, we prove that if a 3-uniform linear-cycle-free hypergraph H, doesn't contain K_5^3 as a subhypergraph, then it is 2-colorable. This result clearly implies that $\alpha \geq \lceil \frac{n}{2} \rceil$. We show that this bound is sharp.

Gyárfás, Győri and Simonovits also proved that a linear-cycle-free 3-uniform hypergraph contains a vertex of strong degree at most 2. In this context, we show that a linear-cycle-free 3-uniform hypergraph has a vertex of degree at most $n-2$ when $n \geq 10$.

1 Introduction

A hypergraph $H = (V, E)$ is k colorable if there is a coloring of the vertices of H with k colors such that there is no monochromatic hyperedge in H. Throughout the paper, we mostly use the terminology introduced in [3].

Definition 1 *A* linear tree *is a hypergraph obtained from a vertex by repeatedly adding hyperedges that intersect the previous hypergraph in exactly one vertex. A* linear path *is a linear tree built so that the next hyperedge always intersects the previous hyperedge in a vertex of degree one.*

A linear cycle *is obtained from a linear path of at least two edges, by adding an edge that intersects the first and the last edges of the linear path in one of their degree one vertices.*

A skeleton T *in* H *is a linear subtree of* H *which cannot be extended to a larger linear subtree by adding a hyperedge* e *of* H *for which* $|e \cap V(T)| = 1$.

An independent set in H is a set of vertices containing no hyperedge of H. More precisely, if I is an independent set of H, then there is no $e \in E(H)$ such that $e \subseteq I$. Let $\alpha(H)$ denote the size of the largest independent set in H. Gyárfás, Győri and Simonovits [3] initiated the study of linear-cycle-free hypergraphs by showing:

Theorem 2 *(Gyárfás, Győri, Simonovits [3]) If* H *is a* 3*-uniform hypergraph on* n *vertices without linear cycles, then it is* 3*-colorable. Moreover,* $\alpha(H) \geq \frac{2n}{5}$.

If the hypergraph does not contain the complete 3-uniform hypergraph K_5^3 as a subhypergraph then a stronger theorem can be proved, answering a question of Gyárfás, Győri and Simonovits.

Theorem 3 *If a 3-uniform linear-cycle-free hypergraph H doesn't contain K_5^3 as a subhypergraph, then it is 2-colorable.*

Corollary 4 *If a 3-uniform linear-cycle-free hypergraph H on n vertices doesn't contain K_5^3 as a subhypergraph, then $\alpha(H) \geq \lceil \frac{n}{2} \rceil$ and the bound is sharp.*

Indeed, from Theorem 3, it trivially follows that $\alpha(H) \geq \lceil \frac{n}{2} \rceil$. The hypergraph H_n on n vertices obtained from the following construction shows that this inequality is sharp. Let H_3 be the hypergraph on 3 vertices v_1, v_2, v_3 such that $v_1v_2v_3 \in E(H_3)$ and let H_4 be the complete 3-uniform hypergraph K_4^3 on 4 vertices v_1, v_2, v_3, v_4. Now for each $3 \leq i \leq n-2$ let us define the hypergraph H_{i+2} such that $V(H_{i+2}) := V(H_i) \cup \{v_{i+1}, v_{i+2}\}$ and $E(H_{i+2}) := E(H_i) \cup \{v_{i+1}v_{i+2}v_j\}_{j=1}^{i}$. If n is even, we start this iterative process with the hypergraph H_4 and if n is odd, we start with H_3. Notice that $\alpha(H_{i+2}) = \alpha(H_i) + 1$ for each i, which implies that $\alpha(H_n) = \lceil \frac{n}{2} \rceil$.

It is another natural problem to bound the number of hyperedges or different types of degrees of vertices in hypergraphs with no linear cycles. The most plausible is the *degree* of a vertex $v \in V$ what is simply the number of hyperedges of H containing v. Given a 3-uniform hypergraph H and $v \in V(H)$, the *link* of v in H is the graph with vertex set $V(H)$ and edge set $\{xy : vxy \in E(H)\}$. The *strong degree* $d^+(v)$ of $v \in V(H)$ is the maximum number of independent edges in the link of v. It is interesting and known for many years that the maximal number of hyperedges in a 3-uniform hypergraph without linear cycles is $\binom{n-1}{2}$, which is the maximum number of hyperedges without a linear triangle [1, 2]. The relation to the strong degree is proved recently.

Theorem 5 *(Gyárfás, Győri, Simonovits [3]) Let H be a 3-uniform hypergraph without linear cycles. Then, it has a vertex v whose strong degree $d^+(v)$ is at most 2.*

In this paper, we show a similar and perhaps more natural theorem concerning the degree of a linear-cycle-free hypergraph.

Theorem 6 *Let H be a 3-uniform hypergraph on $n \geq 10$ vertices without linear cycles. Then, there is a vertex whose degree is at most $n-2$.*

Remark 7 *There is a 3-uniform hypergraph on 9 vertices without linear cycles where the degree of every vertex is 8. This hypergraph H is defined by taking a copy of K_4^3 on vertices $\{u_1, u_2, v_1, v_2\}$ and a vertex disjoint copy of K_5^3 such that $u_1u_2x, v_1v_2x \in E(H)$ for each $x \in V(K_5^3)$ and there are no other hyperedges in H.*

Remark 8 *Theorem 6 cannot be improved because there is a 3-uniform hypergraph H', with $E(H') := \{xab \mid a, b \in V(H') \setminus \{x\}\}$ for a fixed vertex $x \in V(H)$, in which every vertex has degree at least $n-2$.*

Actually, the study of linear cycle free hypergraphs was motivated by Gyárfás and Sárközy, who were motivated by a well-known theorem of Pósa stating that the vertex set of every graph G can be partitioned into at most $\alpha(G)$ cycles where $\alpha(G)$ denotes the independence number of G (where a vertex or an edge is accepted as a cycle).

Gyárfás and Sárközy [4] conjectured that the following extension of Pósa's theorem holds: One can partition every k-uniform hypergraph H into at most $\alpha(H)$ linear cycles (here, as in Pósa's theorem, vertices and subsets of hyperedges are accepted as linear cycles). We showed the following:

Theorem 9 *If H is a 3-uniform hypergraph, then its vertex set can be covered by at most $\alpha(H)$ edge-disjoint linear cycles (where we accept a single vertex or a hyperedge as a linear cycle).*

2 Open questions

The following problems asked by Gyárfás, Győri and Simonovits remain open.

Problem 10 *Can one describe the structure of 3-uniform hypergraphs with no linear cycles?*

It is conceivable that one might construct a linear-cycle-free hypergraph by repeatedly adding hyperedges in a certain fashion. For example, if H is a linear-cycle-free hypergraph, then adding two new vertices u, v to $V(H)$ and adding all the hyperedges of the type uvx for $x \in V(H)$ to $E(H)$, will give us another linear-cycle-free hypergraph.

Problem 11 *Which results extend to r-uniform hypergraphs?*

For $r = 4$ the structure of the "skeleton" seems to be more complicated. It is, however, conceivable that the current methods are useful for this case. In general, the approach of using skeletons seems to be very effective in proving results about linear-cycle-free hypergraphs. It would be interesting to discover more applications of this approach.

And the original version of the conjecture of Gyárfás and Sárközy [4] is still open:

Conjecture 12 *One can partition every k-uniform hypergraph H into at most $\alpha(H)$ linear cycles (here, as in Pósa's theorem, vertices and subsets of hyperedges are accepted as linear cycles).*

Acknowledgment

The research of the second and third authors is partially supported by the National Research, Development and Innovation Office – NKFIH, grant K116769.

References

[1] R. Csákány and J. Kahn. A homological approach to two problems on finite sets, *Journal of Algebraic Combinatorics* **9**, (1999), 141–149.

[2] Z. Füredi and P. Frankl. Exact solution of some Turán-type problems, *Journal of Combinatorial theory A.*, **45** (1987), 226–262.

[3] A. Gyárfás, E. Győri, and M. Simonovits. "On 3-uniform hypergraphs without linear cycles." Journal of Combinatorics **7** (2016), 205–216.

[4] A. Gyárfás and G. Sárközy "Monochromatic loose-cycle partitions in hypergraphs." The Electronic Journal of Combinatorics 21.2 (2014), P2-36.

[5] L. Pósa "On the circuits of finite graphs." Magyar Tud. Akad. Mat. Kutató Int. Közl. 8 (1963), 355-361.

List colourings with restricted lists

Tamás Fleiner[1]

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
Magyar Tudósok körútja 2, Budapest, H-1117.
fleiner@cs.bme.hu

Abstract: We prove an extension of Galvin's theorem, namely that any graph is χ'-edge-choosable if no odd cycle has a common colour in the lists of its edges.

Keywords: list edge-colouring of graphs, list colouring conjecture, stable matchings

1 Introduction

Let $G = (V, E)$ be a finite loopless graph. For each edge $e \in E$, let $L(e) \subset \mathbb{N}$ be a set of available colours for e. We say that G is *L-edge-choosable* if G has an *L-edge-colouring*, that is, a proper edge-coloring $c : E \to \mathbb{N}$ such that $c(e) \in L(e)$ holds for each edge e of E. Graph G is called *k-edge-choosable* if G is L-edge-choosable for any $L : E \to \binom{\mathbb{N}}{k}$. The famous list colouring conjecture states that any finite loopless graph G is $\chi'(G)$-edge-choosable, where chromatic index $\chi'(G)$ denotes the minimum number of colours needed to properly colour the edges of G. By proving the Dinitz conjecture in [2], Galvin essentially justified the list colouring conjecture for (complete) bipartite graphs. In this note, we prove that G is L-edge-choosable whenever $L : E \to \binom{\mathbb{N}}{\chi'(G)}$ and $L^{-1}(i)$ is bipartite for each colour i, that is, if the edges of no odd cycle of G contain a common colour in their lists. Our main tool to achieve this goal is an extension of Galvin's method. Unlike Galvin, here we shall lean on the terminology of stable matchings.

Assume that $G = (V, E)$ is a loopless finite graph and for each vertex v of V, a linear order $\preceq_v$ on the set $E(v)$ of edges incident to v is given. A *matching* of G is a set M of disjoint edges of G and matching M is *stable* if for each edge e of G, there is a vertex v and an edge m of M such that $m \preceq_v e$ holds. The well-known stable marriage theorem states the following.

Theorem 1 (Gale-Shapley [1]) *If $G = (V, E)$ is a finite bipartite graph and $\preceq_v$ is a linear order on $E(v)$ for each vertex v of G then there is a stable matching of G.* □

2 Main result

Our main result is the following.

Theorem 2 *Let $G = (V, E)$ be a finite loopless graph and $c : E \to \{1, 2, \ldots, k\}$ be a proper edge-colouring of G. If $L(e) \subset N$ is a list of at least k colours for each edge e of G and $\bigcap\{L(e) : e \in C\} = \emptyset$ holds for each odd cycle C of G then G is L-edge-choosable.*

[1]Research was supported by MTA-ELTE Egerváry Research Group and the K108383 OTKA grant.

PROOF: For $i = 1, 2, \ldots$ define $E_i := \{e \in E : 2i - 1 \leq c(e) \leq 2i\}$. Clearly, $E = E_1 \cup E_2 \cup \ldots \cup E_{\lceil k/2 \rceil}$. As the maximum degree in $G_i = (V, E_i)$ is not more than 2, each component of G_i is a path or a cycle. Orient the edges of G such that each component of each G_i becomes a directed path or a directed cycle. For edge $e = uv \in E_i$ define

$$r(e, v) = \begin{cases} i & \text{if } v \text{ is the head of the arc that corresponds to } e \\ k + 1 - i & \text{if } v \text{ is the tail of the arc that corresponds to } e \end{cases}$$

Observe that if $r(e, v) = r(f, v)$ then e and f must belong to the same set E_i and orientations of e and f either both enter or both leave v. Hence $r(e, v) = r(f, v)$ implies $e = f$ and consequently $\preceq_v$ is a linear order on $E(v)$ where $e \preceq_v f$ means that $r(e, v) \leq r(f, v)$. Assume now that $e = uv$ is the oriented version of edge $e \in E_i$. From $r(e, u) = i$ and $r(e, v) = k + 1 - i$ we get that

$$|\{f \in E(u) : f \preceq_u e\}| + |\{f \in E(v) : f \preceq_v e\}| \leq i - 1 + (k + 1 - i) - 1 = k - 1 \ . \tag{1}$$

The above observations enable us to employ Galvin's method to finish the proof. Define $E^i := \{e \in E : i \in L(e)\}$ as the set of i-colourable edges and let $G^i := (V, E^i)$. As none of the G^is contain an odd cycle by the assumption, each G^i is bipartite. For $i = 0, 1, 2, \ldots$ define M^i as a stable matching of graph $G^i \setminus (M^0 \cup \ldots \cup M^{i-1})$ with restricted linear orders $\preceq_v$. Such a matching exists by Theorem 1.

To show that G is L-edge-choosable, give colour i to edges of M^i. Clearly, no two edges of the same colour share a vertex and each coloured edge receives its colour from its list. The only thing left is to show that each edge of G receives some colour.

Observe that if edge $e = uv$ of G^i does not receive colour i, (i.e. if $e \notin M^i$) then either $e \in M^j$ for some $j < i$ (hence e received colour j before M^i was defined) or M^i contains an edge f such that $f \preceq_u e$ or $f \preceq_v e$. So if e does not receive any colour, that is, if $e \notin \bigcup\{M^j : j \in L(e)\}$ then there is an $f^j \in M^j$ for each $j \in L(e)$ with $f^j \preceq_u e$ or $f^j \preceq_v e$. As $|L(e)| \geq k$, this is impossible by (1) and this contradiction proves that the above algorithm finds a proper L-edge-colouring of G. $\square$

References

[1] D. Gale and L.S. Shapley. College admissions and stability of marriage. *Amer. Math. Monthly*, 69(1):9–15, 1962.

[2] Fred Galvin. The list chromatic index of a bipartite multigraph. *J. Combin. Theory Ser. B*, 63(1):153–158, 1995.

On packing spanning arborescences with matroid constraint

Quentin Fortier

Univ. Grenoble Alpes
G-SCOP
46, Avenue Félix Viallet
Grenoble, France, 38000
quentin.fortier@grenoble-inp.fr

Csaba Király

Department of Operations Research
Eötvös Loránd University
Pázmány Péter sétány 1/C
Budapest, Hungary, 1117
cskiraly@cs.elte.hu

Zoltán Szigeti

Univ. Grenoble Alpes
G-SCOP
46, Avenue Félix Viallet
Grenoble, France, 38000
zoltan.szigeti@grenoble-inp.fr

Shin-ichi Tanigawa

RIMS, Kyoto University
Sakyo-ku, Kyoto 606-8502, Japan
and
Centrum Wiskunde & Informatica (CWI)
Postbus 94079, 1090 GB
Amsterdam, The Netherlands
tanigawa@kurims.kyoto-u.ac.jp

Abstract: Let us be given a rooted digraph $D = (V + s, A)$ with a designated root vertex s. Edmonds' seminal result [4] states that D has a packing of k spanning s-arborescences if and only if D has a packing of k (s,t)-paths for all $t \in V$, where a packing means arc-disjoint subgraphs.

Let $\mathcal{M}$ be a matroid on the set of arcs leaving s. A packing of (s,t)-paths is called $\mathcal{M}$-based if their arcs leaving s form a base of $\mathcal{M}$ while a packing of s-arborescences is called $\mathcal{M}$-based if, for all $t \in V$, the packing of (s,t)-paths provided by the arborescences is $\mathcal{M}$-based. Durand de Gevigney, Nguyen and Szigeti proved in [3] that D has an $\mathcal{M}$-based packing of s-arborescences if and only if D has an $\mathcal{M}$-based packing of (s,t)-paths for all $t \in V$. Bérczi and Frank conjectured that this statement can be strengthen in the sense of Edmonds' theorem such that each s-arborescence is required to be spanning. Specifically, they conjectured that D has an $\mathcal{M}$-based packing of spanning s-arborescences if and only if D has an $\mathcal{M}$-based packing of (s,t)-paths for all $t \in V$.

We disprove this conjecture in its general form and we prove that the corresponding decision problem is NP-complete. However, we prove that the conjecture holds for several fundamental classes of matroids, such as graphic matroids and transversal matroids.

Keywords: connectivity, spanning, arborescence, packing, matroid

1 Introduction

Packing different kinds of objects is a natural question in real life. In optimization problems, the goal is to maximize the number of objects in the packing. A wide variety of problems can be modeled as packing problems, and fundamental problems in combinatorial optimization, such as bin packing, path packing, tree packing, are of this type. This paper deals with packing problems of arborescences, or more

generally, packing problems concerning connectivity in directed graphs. Here, by packing subgraphs in a directed graph, we mean a set of arc-disjoint subgraphs.

The question of reachability is one of the basics in the area of connectivity in digraphs. Suppose that we are given a **rooted digraph**, i.e. a digraph $D = (V + s, A)$ with a designated root vertex s. Let S be the set of vertices reachable from s in D. The definition of reachability says that, for each $t \in S$, D has an (s,t)-path, which certificates that t indeed belongs to S. Now, consider storing such certificates for all vertices in S. Then storing an s-arborescence on S would be the most compact way for keeping all the certificates simultaneously.

To extend this idea to a more general setting, suppose that D has a packing of k (s,t)-paths from s to each vertex t in V, and suppose that we want to provide a certificate that D indeed has such a property. Then the most compact certificate would be to exhibit k arc-disjoint spanning s-arborescences in D. The following fundamental theorem of Edmonds [4] claims that such a compact certificate always exists.

Theorem 1 ([4]) *There exists a packing of k spanning s-arborescences in a rooted digraph $D = (V+s, A)$ if and only if there exists a packing of k (s,t)-paths in D for every $t \in V$.* □

The problem of packing k (s,t)-paths is equivalent to asking whether one can send k distinct commodities from s to t by assuming that each arc can transmit at most one commodity. Then what happens if commodities have a more involved independence structure? Here we are interested in a situation that each arc from the root can be used to transmit only a particular commodity, and we would like to know when every vertex can receive a sufficient amount of independent commodities to understand the whole structure.

More formally, suppose that we are given a **matroid-rooted digraph** $(D = (V + s, A), \mathcal{M})$, i.e. a matroid $\mathcal{M}$ is given on the set of arcs leaving the root s that we call **root arcs**. We are interested in a packing of (s,t)-paths whose root arcs form a base of $\mathcal{M}$. Such a packing is said to be an **$\mathcal{M}$-based packing** of (s,t)-paths. A packing of s-arborescences is called **$\mathcal{M}$-based** if, for all $t \in V$, the packing of (s,t)-paths provided by the arborescences is $\mathcal{M}$-based. A natural question is whether Edmonds' theorem can be extended for $\mathcal{M}$-based packings. A result of Durand de Gevigney, Nguyen and Szigeti [3] gives a partial answer to this question by showing the *equivalence of the existence of an $\mathcal{M}$-based packing of s-arborescences in D and an $\mathcal{M}$-based packing of (s,t)-paths in D for every $t \in V$.*

Notice that at the quantitative level, Theorem 1 always guarantees the existence of k spanning s-arborescences (provided the condition is satisfied) while the number of s-arborescences in the result of [3] may be more than the rank of $\mathcal{M}$ since these arborescences are not necessarily spanning. Bérczi and Frank [9] conjectured that the result of [3] can be strengthen in the sense of Edmonds' theorem. This conjecture appeared also in a paper of Bérczi, T. Király and Kobayashi [2]. More formally, the conjecture is the following.

Conjecture 2 ([2]) *Let $(D = (V + s, A), \mathcal{M})$ be a matroid-rooted digraph. There exists an $\mathcal{M}$-based packing of **spanning** s-arborescences in D if and only if there exists an $\mathcal{M}$-based packing of (s,t)-paths in D for every $t \in V$.*

Contributions. We will prove that Conjecture 2 is true for several fundamental classes of matroids such as graphic and transversal matroids. The main result of this paper is that Conjecture 2 is false in its general form. We will even prove that the following decision problem is NP-complete.

Problem 3 *Given a matroid-rooted digraph $(D = (V+s, A), \mathcal{M})$, decide whether there exists an $\mathcal{M}$-based packing of spanning s-arborescences in D.*

Key ideas. We present the main ideas of the proofs below. More details are given in Sec. 2 and 3 while the full proofs can be found in [6].

Graphic matroids. Let $(D, \mathcal{M})$ be a matroid-rooted digraph where $\mathcal{M}$ is a graphic matroid of rank k. Let $G = (\{0, 1, \ldots, k\}, E)$ be a connected undirected graph representing $\mathcal{M}$, so the edges of G corresponds to the root arcs of D. The idea is to require for the packing that each root arc may belong to T_i only

if its corresponding edge is incident to $i \in \{0, 1, \ldots, k\}$ in G. This condition gives an extra property for the packing obtained by induction, based on which we show how to extend the packing while keeping it $\mathcal{M}$-based.

Transversal matroids. Let $(D, \mathcal{M})$ be a matroid-rooted digraph where $\mathcal{M}$ is a transversal matroid of rank k. Let $G = (S, T; E)$ be a bipartite graph representing $\mathcal{M}$ where S corresponds to the set of root arcs of D and $T = \{1, \ldots, k\}$. The plan is to replace the matroid-based condition by the following new condition: a root arc may belong to T_i only if its corresponding vertex is connected to i in G. It is much easier to deal with this condition, and the key observation is that if a packing of arborescences satisfies this new condition then any set of k root arcs belonging to different arborescences of the packing forms a base of $\mathcal{M}$. Thus the packing is automatically $\mathcal{M}$-based.

Counterexample and NP-completeness. One of the simplest non-graphic and non-transversal matroids is the Fano matroid. A simple proof shows that Conjecture 2 is true for the Fano matroid in the special case where the digraph is acyclic. However, it turns out that Conjecture 2 is false (also in this special case) when we allow to extend the Fano matroid by parallel elements. The symmetry of the Fano matroid will be widely explored in the proof, and also its principal property will be important that every pair of its elements is contained in a dependent set of cardinality 3, i.e. in a line of the Fano plane. For both results, we will construct our acyclic digraphs step by step by adding sink vertices of in-degree 3. This construction will ensure not only the existence of the required $\mathcal{M}$-based path packings but also that every possible $\mathcal{M}$-based arborescence packing is an extension of the previous instance. We design each construction step so that possible extensions are restricted.

Related works. Connectivity is one of the most well-studied properties of graphs. The earliest results related to our main interest on packing problems concerning connectivity are the papers of Nash-Williams [17] and Tutte [20] on packing trees in undirected graphs from 1961. The topic of packing arborescences has been extensively studied in the seventies by Edmonds and Frank [4, 7]. The connection between these problems was pointed out in a work of Frank [8] on orientations of graphs.

The hypergraphic counterparts of the above packing results were discovered by Frank, T. Király, Z. Király and Kriesell [10, 11]. A surprising extension of Edmonds' result was given by Katoh, Kamiyama and Takizawa [13] and Fujishige [12] for the case when no spanning arborescences exist. Szegő [19] gave an abstract version of Edmonds' result that was extended to an abstract version of the result of [13] in a paper of Bérczi and Frank [1].

Investigations in rigidity theory inspired an extensive research on possible extensions of Nash-Williams' and Tutte's result. Katoh and Tanigawa [14] generalized this tree packing result for the problem of "matroid-based packing of rooted trees" and presented several applications of this result in rigidity theory. Durand de Gevigney, Nguyen and Szigeti [3] used the techniques of Frank to show that, by an extension of Edmonds' result, an alternative proof of the packing result of [14] can be obtained. These breakthrough results inspired an intensive research in the last few years on this topic to extend the above mentioned results, see [2, 5, 15, 16].

Definitions. An ***s*-arborescence** is a directed tree on a vertex-set containing the **root** vertex s in which each vertex has in-degree 1 except s. An s-arborescence in a digraph $D = (V + s, A)$ is **spanning** if its vertex set is $V + s$. For an s-arborescence T and a vertex $v \neq s$ of T, we denote the unique arc of T entering v by $\boldsymbol{T(v)}$, the unique path from s to v by $\boldsymbol{T[s, v]}$, and its first arc by $\boldsymbol{e_{T[s,v]}}$. For disjoint sets $X, Y \subseteq V + s$, we denote by $\boldsymbol{\partial^D_X(Y)}$ the subset of arcs in D with tail in X and head in Y. The superscript D will be omitted, when it is clear from the context. The in-degree of a set $X \subseteq V + s$ is denoted by $\boldsymbol{\varrho_D(X)} := |\partial^D_{V+s-X}(X)|$.

We will use standard terminology from matroid theory, such as rank functions, independent sets, and bases. For details, we refer to [18]. We usually denote a matroid $\mathcal{M}$ by a pair (S, r) of the ground set S and the rank function $r : 2^{\mathsf{S}} \to \mathbb{Z}$. We define $\mathbf{Span(Q)} := \{\mathsf{s} \in \mathsf{S} : r(\mathsf{Q} \cup \{\mathsf{s}\}) = r(\mathsf{Q})\}$. Note that $\mathrm{Span}_{\mathcal{M}}$ is monotone. Two elements $\mathsf{a}, \mathsf{a}' \in \mathsf{S}$ are said to be **parallel** in $\mathcal{M} = (\mathsf{S}, r)$ (in notation, $\mathbf{a} \parallel \mathbf{a'}$) if $r(\{\mathsf{a}\}) = r(\{\mathsf{a}'\}) = r(\{\mathsf{a}, \mathsf{a}'\}) = 1$.

We say that a matroid-rooted digraph $(D = (V+s, A), \mathcal{M} = (\partial_s(V), r))$ is **rooted $\mathcal{M}$-arc-connected** ($\mathcal{M}$**-ac** for short) if there exists an $\mathcal{M}$-based packing of (s, t)-paths for all vertices t in V. One can easily

prove a Menger type theorem saying that D is rooted $\mathcal{M}$-arc-connected if and only if

$$r(\partial_s(X)) + \varrho_{D-s}(X) \geq r(\mathcal{M}) \text{ for all } X \subseteq V, \tag{1}$$

where $\boldsymbol{r(\mathcal{M})}$ denotes the rank of $\mathcal{M}$. For simplicity, we will call an $\mathcal{M}$-based packing of spanning s-arborescences in D that covers $\partial_s(V)$ a **feasible packing**.

The following classes of matroids will be discussed in this paper. Given a graph $G = (V, E)$ with a bijection $\pi : E \to \mathsf{S}$, a matroid on S with independent sets in $\mathcal{I} := \{\pi(F) : F \text{ is the edge set of a forest of } G\}$ is called a **graphic matroid**. A **Fano matroid** is a rank 3 matroid derived from the Fano plane (the smallest projective plane with 7 points) on a 7 element ground set (the points of the Fano plane) where every set of cardinality 3 is a base except the lines of the Fano plane. Given a bipartite graph $G = (S, T; E)$ with a bijection $\pi : S \to \mathsf{S}$, a matroid on S with independent sets in $\mathcal{I} := \{\pi(X) : X \subseteq S$ that can be covered by a matching in $G\}$ is called a **transversal matroid**. A special class of transversal matroids where G is the complete bipartite graph $K_{n,k}$ is called the **uniform matroid $\boldsymbol{U_{k,n}}$**. It is well-known that a graphic matroid is always representable by a connected graph and a transversal matroid is always representable by a bipartite graph where $|T|$ is equal to the rank. It is also well known that a matroid of rank at most 3 is not graphic if and only if it has a "minor" isomorphic to the Fano matroid or $U_{2,4}$.

2 Positive results

In this section, we prove Conjecture 2 for several special cases. The necessity of Conjecture 2 is always true and is easy to prove, so we will only prove the sufficiency in each case.

Some of our positive results are obtained by extending the proof given by [3], and hence we shall first review it by introducing several key ingredients used later. In [3], the result was proved in a slightly stronger form as stated in our introduction by imposing an extra technical condition as follows. Let $(D = (V + s, A), \mathcal{M})$ be a matroid-rooted digraph. D is called **$\mathcal{M}$-independent** if $\partial_s(v)$ is independent in $\mathcal{M}$ for every $v \in V$. This condition ensures that each root arc can be used in an $\mathcal{M}$-based packing of s-arborescences in D, as follows.

Theorem 4 ([3]) *Let $(D = (V + s, A), \mathcal{M} = (\partial_s(V), r))$ be a matroid-rooted digraph. There exists an $\mathcal{M}$-based packing of s-arborescences in D that covers $\partial_s(V)$ if and only if D is $\mathcal{M}$-ac and $\mathcal{M}$-independent.*

Observe that, by omitting some root arcs of an $\mathcal{M}$-ac digraph, one can get an $\mathcal{M}'$-ac and $\mathcal{M}'$-independent digraph where $\mathcal{M}'$ is a submatroid of $\mathcal{M}$ with the same rank. Hence this result is indeed a bit stronger. Observe also that $\mathcal{M}$-independence is a trivial necessary condition for an $\mathcal{M}$-based packing covering $\partial_s(V)$.

Let $(D, \mathcal{M})$ be as in Theorem 4. We call $X \subseteq V$ **tight** if (1) holds with equality. We say that a non-root arc uv is **good** if $\partial_s(u) \not\subseteq \mathrm{Span}_{\mathcal{M}}(\partial_s(v))$. A pair (uv, x) of a good arc uv in $D - s$ and $x \in \partial_s(u) - \mathrm{Span}_{\mathcal{M}}(\partial_s(v))$ is said to be **admissible** if there is no tight set X with $v \in X$ and $u \notin X$ such that x is in the span of $\partial_s(X)$. The **shifting** (of $(D, \mathcal{M})$) along (uv, x) is a new instance $(D', \mathcal{M}')$ obtained from $(D, \mathcal{M})$ by removing uv and inserting a new root arc sv such that sv is a parallel element to x in the underlying matroid. Note that shifting satisfies $\mathcal{M}$-independence (resp. rooted $\mathcal{M}$-arc-connectivity) if and only if uv is good (resp., (uv, x) is admissible).

The proof of the sufficiency of Theorem 4 is done by induction on the number of non-root arcs. If no good arc exists, then the set of root arcs form an $\mathcal{M}$-based packing of s-arborescences. Otherwise, it is proved in [3] that there exists an admissible pair (e, x), and hence the shifting $(D', \mathcal{M}')$ along (e, x) is $\mathcal{M}'$-independent and $\mathcal{M}'$-ac. By induction, there exists an $\mathcal{M}'$-based packing $\mathcal{T}$ of s-arborescences in D' such that it covers $\partial'_s(V)$. We can suppose that each s-arborescence in $\mathcal{T}$ has only one root arc since otherwise we can split it into several s-arborescences to satisfy this condition. Let $T \in \mathcal{T}$ be the arborescence covering x and $T' \in \mathcal{T}$ the arborescence covering the new root arc f in D'. Then $(\mathcal{T} - \{T, T'\}) \cup \{T \cup (T' - f) + e\}$ is a desired $\mathcal{M}$-based packing of s-arborescences in D that covers $\partial_s(V)$, and this completes the proof of Theorem 4. □

Now consider applying the proof to Conjecture 2. In the same manner, by induction, one gets an $\mathcal{M}'$-based packing $\mathcal{T}$ of *spanning* s-arborescences in D'. Our goal is to construct a feasible packing in D based on $\mathcal{T}$. Let $T \in \mathcal{T}$ be an arborescence that covers the new root arc f of D'. If T also contains x, then $(\mathcal{T} - \{T\}) \cup \{T - f + e\}$ is a feasible packing in D, and we are done. The difficult case is when T does not contain x. We will show how to overcome this difficulty by new ideas if $\mathcal{M}$ has rank at most 2 or is graphic.

In the case where the matroid has rank at most 2, the previous proof fails only when the packing consists of two arborescences T_1 and T_2 (thus the rank of $\mathcal{M}'$ is 2), w.l.o.g. assume $x \in T_1$ and $f \in T_2$. Let $V_f \subseteq V$ be the set of vertices which is reachable from s in T_2 by a path starting with the arc f or an arc parallel to f in $\mathcal{M}$. Let $\{T_1^*, T_2^*\}$ be the packing that arises from $\{T_1, T_2\}$ by exchanging the arcs $T_1(v)$ and $T_2(v)$ for every vertex v in V_f. Then we can prove that $\{T_1^*, T_2^*\}$ is a feasible packing in D' where x and f are in T_1^*. Thus we are in a case already treated. Therefore, we get the following theorem.

Theorem 5 *Let $(D = (V + s, A), \mathcal{M} = (\partial_s(V), r))$ be a matroid-rooted digraph with $r(\mathcal{M}) \leq 2$. There exists an $\mathcal{M}$-based packing of spanning s-arborescences in D that covers $\partial_s(V)$ if and only if D is $\mathcal{M}$-independent and $\mathcal{M}$-ac.* □

Now we turn to the proof of Conjecture 2 for graphic matroids.

Theorem 6 *Let $(D = (V + s, A), \mathcal{M})$ be a matroid-rooted digraph where $\mathcal{M} = (\partial_s(V), r)$ is a graphic matroid of rank k. There exists an $\mathcal{M}$-based packing of spanning s-arborescences in D covering $\partial_s(V)$ if and only if D is $\mathcal{M}$-ac and $\mathcal{M}$-independent.*

PROOF: Let $\boldsymbol{G = (\{0, 1, \ldots, k\}, E)}$ be a connected undirected graph with a bijection $\boldsymbol{\pi}$: $E \to \partial_s(V)$ representing $\mathcal{M}$. We will refer to $(D, \mathcal{M})$ as (D, G, π). For $e \in E$, let $\boldsymbol{x_e} = \pi(e)$. For $X \subseteq V$, let $\boldsymbol{E_X} = \pi^{-1}(\partial_s(X))$ and $\boldsymbol{C_X}$ the vertex set of the connected component $\boldsymbol{Q_X}$ of $(V(G), E_X)$ that contains 0. Let $v \in V$. As D is $\mathcal{M}$-independent, $k \geq |E_v|$ and Q_v is a tree. Let $\boldsymbol{\overrightarrow{Q}_v}$ be the 0-arborescence that arises by orienting each edge e of Q_v to $\boldsymbol{\overrightarrow{e}}$.

We impose the following extra property for the packing $\{T_1, \ldots, T_k\}$:

$$\text{for } \overrightarrow{e} = ij \text{ belonging to } \overrightarrow{Q}_v \text{ for some } v \in V, \ x_e \text{ belongs to } T_j. \tag{2}$$

Let (D, G, π) be a counterexample minimizing $\sum_{v \in V}(k - |E_v|) \geq 0$. Let $\boldsymbol{v^*} \in V$ such that $|C_{v^*}|$ is as small as possible. If $C_{v^*} = V(G)$, then Q_v is a spanning tree of G for every $v \in V$. In this case, using only the root arcs, the 0-arborescences $\overrightarrow{Q}_v$ show how to define a feasible packing satisfying (2).

From now on, we suppose that $C_{v^*} \subsetneq V(G)$. Let $\boldsymbol{W} = \{v \in V : C_v = C_{v^*}\}$. Then $C_W = C_{v^*}$. For $p \in V - W$, an element $e \in E_p$ is called **critical** if $\overrightarrow{e}$ belongs to $\overrightarrow{Q}_p$ and $\overrightarrow{e}$ leaves C_W. By the minimality of $|C_{v^*}|$ and $p \in V - W$, we have $C_p - C_W \neq \emptyset$. Hence the following claim follows from the fact that $\overrightarrow{Q}_p$ is a spanning 0-arborescence on C_p.

Claim 7 *For $p \in V - W$, E_p contains a critical element.*

For a critical element e, if (pq, x_e) is admissible, then by (2) one can construct a feasible packing of D from that of the shifting along (pq, x_e), contradicting that $(D, \mathcal{M})$ is a counterexample. Thus we have the following.

Claim 8 *Let $pq \in \partial_{V-W}(W)$ and $e \in E_p$ critical. Then (pq, x_e) is not admissible.*

By $C_W \subsetneq V(G)$, $r(\pi(E_W)) < k$. Therefore, by (1), D has an arc pq with $p \in V - W$ and $q \in W$. By Claim 7, E_p contains a critical element e, and by Claim 8 (pq, x_e) is not admissible. In other words, there exists a tight set $X \subseteq V$ with $q \in X$, $p \notin X$ and $x_e \in \mathrm{Span}(\pi(E_X))$.

Let (pq, x_e) be such a pair so that X is minimal. As e is critical, $x_e \in \mathrm{Span}(\pi(E_X)) - \mathrm{Span}(\pi(E_W))$. Hence $r(\pi(E_{X \cap W})) < r(\pi(E_X))$. By (1) and the tightness of X, $\varrho_{D-s}(X \cap W) \geq k - r(\pi(E_{X \cap W})) >$

$k-r(\pi(E_X)) = \varrho_{D-s}(X)$. Hence $D-s$ has an arc $p'q'$ with $p' \in X-W$ and $q' \in X \cap W$. Since $E_{p'}$ contains a critical element e' by Claim 7, $(p'q', x_{e'})$ is not admissible by Claim 8, that is, there exists a tight set $X' \subseteq V$ with $q' \in X'$ and $p' \notin X'$ such that $x_{e'} \in \mathrm{Span}(\pi(E_{X'}))$. Since $p' \in X-W$, $E_{p'} \subseteq E_X$ and hence $e' \in E_X$. By [3, Claim 2.1(a)], $X \cap X'$ is tight and $x_{e'} \in \mathrm{Span}(\pi(E_{X \cap X'}))$. Furthermore, $q' \in X \cap X'$, $p' \notin X \cap X'$, and $e' \in E_{p'}$ is critical, contradicting the minimal choice of X, since $p' \in X - X'$. □

The case when $\mathcal{M}$ is transversal can be solved by a completely different idea, by reducing the problem to a packing problem of reachability branchings. For a non-empty set $R \subseteq U$, the subdigraph $T = (U, A')$ of a digraph $D^* = (V^*, A)$ is said to be an **R-branching** if it consists of $|R|$ vertex-disjoint arborescences in D^* whose roots are in R. We say that T is a **reachability R-branching** in D^* if U is the set of reachable vertices from a vertex in R in D^*. The following surprising generalization of Edmonds' theorem was discovered by Kamiyama, Katoh and Takizawa [13].

Theorem 9 ([13]) *Let $D^* = (V^*, A)$ be a digraph and $\mathcal{R} := \{R_1, ..., R_k\}$ a family of non-empty subsets of V^*. There exits a packing of reachability $\mathcal{R}$-branchings in D^* if and only if $\varrho_{D^*}(X) \geq p_{\mathcal{R}}(X)$ for every $\emptyset \neq X \subseteq V^*$ where $p_{\mathcal{R}}(X)$ denotes the number of R_i's for which $R_i \cap X = \emptyset$ and there exits a path from a vertex in R_i to a vertex in X.* □

We prove now that Conjecture 2 is true for transversal matroids.

Theorem 10 *Let $(D = (V+s, A), \mathcal{M} = (\partial_s(V), r))$ be a matroid-rooted digraph, where $\mathcal{M}$ is a transversal matroid. There exists an $\mathcal{M}$-based packing of spanning s-arborescences in D if and only if D is $\mathcal{M}$-ac.*

Proof: Let $G = (S, T; E)$ be a bipartite graph representing $\mathcal{M}$ such that $T = \{1, \ldots, k\}$, where $k = r(\mathcal{M})$, and $\pi : S \to \partial_s(V)$ a bijection. Let $D^* = (V^*, A^*)$ be the digraph that arises from D by splitting s into $|S|$ new vertices of out-degree one. Let r_e denote the tail of e in D^* for each $e \in \partial_s^D(V)$, R^* the set of new vertices r_e and $R_i = \{r_e \in R^* : \pi^{-1}(e)$ is adjacent to i in $G\}$ for $i \in T$.

It can be proved that every vertex $v \in V^* - R^* (= V - s)$ is reachable from each R_i in D^* and condition of Theorem 9 holds. Thus, by Theorem 9, there exists a packing of reachability $\{R_1, \ldots, R_k\}$-branchings in D^* where each reachability R_i-branching B_i covers $V - r$. By contracting R^* into s, we obtain k pairwise arc-disjoint spanning s-arborescences $T_i = B_i/R^*$ in D. The construction implies that, for each root arc e in T_i, G has an edge between $\pi^{-1}(e)$ and i. Therefore, for each $v \in V$, for each $i \in \{1, \ldots, k\}$ and for the root arc e in $T_i[s, v]$, $\pi^{-1}(e)$ is connected to i in G implying that these root arcs over all i form a base of $\mathcal{M}$. Hence $T_1, \ldots, T_k$ indeed form a feasible packing. □

3 Negative results

We will give a counterexample to Conjecture 2 and prove that Problem 3 is NP-complete for acyclic digraphs and a certain class of matroids as follows.

Theorem 11 *There exist an acyclic digraph $D = (V + s, A)$ and a matroid $\mathcal{M}$ of rank three such that $(D, \mathcal{M})$ is a counterexample to Conjecture 2.*

Theorem 12 *Problem 3 is NP-complete even if $D = (V + s, A)$ is acyclic and $\mathcal{M}$ is a linear matroid of rank three with a given linear representation.*

In our constructions, D will always be acyclic and $\mathcal{M}$-independent. Hence the condition (1) can be significantly simplified to

$$\varrho_D(v) \geq r(\mathcal{M}) \text{ for all } v \in V. \tag{3}$$

As we noted before, the matroid $\mathcal{M}$ used in the constructions, that we call a **parallel extension of the Fano matroid**, will arise from the Fano matroid by adding some parallel copies of its elements.

The proofs are done by defining several gadget constructions, each of which restricts possible packings. Each construction step is referred to as an **operation** below, and we shall define several distinct operations. In each construction, we insert new vertices one by one together with three new arcs entering it. A new root arc will always be added keeping the $\mathcal{M}$-independence as well as the fact that $\mathcal{M}$ is a parallel extension of the Fano matroid (or its submatroid). Thus, $D = (V + s, A)$ is always acyclic and, by (3), the resulting instance $(D, \mathcal{M})$ will be $\mathcal{M}$-ac by (3). Hence in the subsequent discussion we omit to mention that $(D, \mathcal{M})$ is $\mathcal{M}$-independent and $\mathcal{M}$-ac.

We say that a vertex $v \in V$ **gets** a base B in a feasible packing $\{T_1, T_2, T_3\}$ if $B = \{e_{T_1[s,v]}, e_{T_2[s,v]}, e_{T_3[s,v]}\}$. We also say that v **gets** $e_{T_i[s,v]}$ **from** u if u is on the path $T_i[s, v]$ $(i = 1, 2, 3)$. T_1, T_2 and T_3 will be called the red, blue and black arborescences, resp. We say that an element of $\mathcal{M}$ is **colored** by λ if it is in the arborescence of color λ. In the following, the elements of $\mathcal{M}$ will be denoted by the first 7 letters of the alphabet and apostrophes will be used when we consider a parallel element of a previously used one (that may be also an identical element to this previous one).

We also remark that, as we will always extend a digraph by adding a vertex of out-degree zero one by one, every feasible packing of the resulting digraph is an extension of a feasible packing of the original digraph. By using the following operations, we shall control possible extensions of packings.

Because the remaining space is not sufficient to describe the full detail of operations, in this extended abstract, we only sketch the idea of each operation and how to combine those operations together. For the precise definition of each operation, see [6, Sec. 4]. First, we mention three operations that are needed for both proofs and then we prove Theorem 11.

Operation 13 *Given $(D, \mathcal{M})$, suppose that $u, v \in V$ get the bases $\{a, b, c\}$ and $\{a', b', c'\}$ in every feasible packing, resp., where $a' \parallel a$, $b' \parallel b$ and $c' \parallel c$.* **Avoid-flip $AF_a(u, v)$** *extends $(D, \mathcal{M})$ to $(D', \mathcal{M}')$ by adding 5 new vertices $w_1, \ldots, w_5$ to D and 4 new elements to $\mathcal{M}$ such that $(D', \mathcal{M}')$ satisfies the following property: every feasible packing in D extends to a feasible packing in D' except those where a and a' have the same color and the colors of the pairs (b, b') and (c, c') are different.*

Operation 14 *Given $(D, \mathcal{M})$, suppose that $u, v \in V$ get the bases $\{a, b, c\}$ and $\{a', b', c'\}$ in every feasible packing, resp., where $a' \parallel a$, $b' \parallel b$ and $c' \parallel c$.* **Copy-one-color $COC_b(u, v)$** *extends $(D, \mathcal{M})$ to $(D', \mathcal{M}')$ by adding 11 new vertices to D and 8 new elements to $\mathcal{M}$ such that $(D', \mathcal{M}')$ has the following property: every feasible packing in D extends to that in D' except those where the colors of b and b' are different.*

Operation 15 *Given $(D, \mathcal{M})$, suppose that $v \in V$ gets the base $\{a, b, c\}$ in every feasible packing.* **Change-colors $CC_{a,c}(v)$** *extends $(D, \mathcal{M})$ to $(D', \mathcal{M}')$ by adding 111 new vertices to D and 114 new elements to $\mathcal{M}$ such that every feasible packing in D extends to a feasible packing in D'. Moreover, $(D', \mathcal{M}')$ has a new vertex w having the following property: if the base (a, b, c) got by v is colored by $(\lambda_1, \lambda_2, \lambda_3)$, then w gets a base (a'', b, c'') colored by $(\lambda_3, \lambda_2, \lambda_1)$.*

We use $w = CC_{a,c}(v)$ to denote the new vertex w given in Operation 15.

Proof of Theorem 11. We start with a digraph on two vertices, a root s and the other vertex v, along with 3 parallel arcs a, b and c from s to v. The underlying matroid is the free matroid on $\partial_s(v)$. We extend this by using the operations defined above. In the following, the arborescences covering a, b and c will be called red, blue and black, resp. By using $CC_{a,b}(v)$, the instance is extended such that $w = CC_{a,b}(v)$ gets a base (a'', b, c'') with elements parallel to the elements of a, b and c and colors $(\text{black}, \text{blue}, \text{red})$. We further extend the instance by $AF_b(v, w)$. Then, no feasible packing exists in the resulting instance. Since the construction keeps (3), the resulting instance is rooted $\mathcal{M}$-arc-connected, and hence is a counterexample to Conjecture 2. □

Now we turn to the proof of Theorem 12. Problem 3 is in NP in the case where a linear representation of the matroid is given as input since the packing itself is a witness for the problem that can be checked in polynomial time. We will use the well-known 3-SAT to prove the NP-completeness of our problem.

Let us take a 3-CNF formula. In order to express each clause, our idea is to represent it as a concatenation of majority functions and implement each majority function by using our operations.

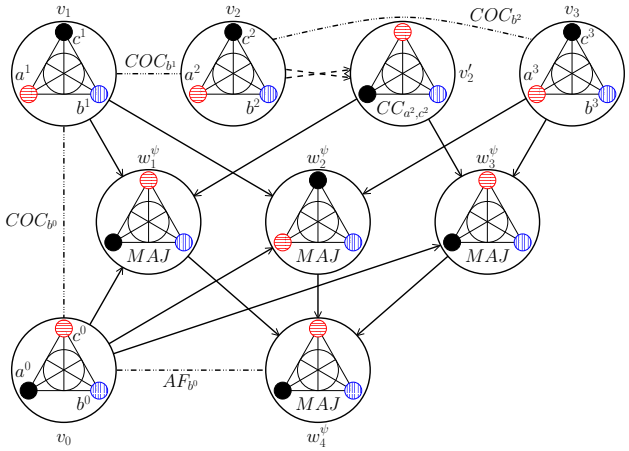


Figure 1: A part of the construction in the proof of Theorem 12. This demonstrates how the assignment $x_1 = x_2 = x_3 = 0$ makes the clause $\psi = x_1 \vee \bar{x}_2 \vee x_3$ true in the corresponding feasible packing. The crossing dashed arcs represent the operation CC.

Recall that the majority function $\mathsf{maj}(\alpha, \beta, \gamma)$ is a Boolean function that has a value 1 if and only if at least two among α, β, γ have value 1. Observe that, given $\alpha, \beta, \gamma \in \{0, 1\}$,

$$\alpha \vee \beta \vee \gamma = \mathsf{maj}(\mathsf{maj}(\alpha, \beta, 1), \mathsf{maj}(\alpha, \gamma, 1), \mathsf{maj}(\beta, \gamma, 1)). \tag{4}$$

Operation 16 *Given $(D, \mathcal{M})$, suppose that $v_1, v_2, v_3 \in V$ get the bases $\{a, b, c\}$, $\{a', b', c'\}$ and $\{a'', b'', c''\}$, resp., in every feasible packing where $a \parallel a' \parallel a''$, $b \parallel b' \parallel b''$ and $c \parallel c' \parallel c''$. Operation* **Majority $MAJ(v_1, v_2, v_3)$** *extends $(D, \mathcal{M})$ to $(D', \mathcal{M}')$ by adding a new vertex w with 3 incoming arcs v_1w, v_2w and v_3w. Consider a feasible packing of D such that all of b, b' and b'' are colored by λ (and hence there are only two types of possible coloring schemes on each v_i). Then the packing extends to a feasible packing of D'. Moreover, in every such extension w gets a base formed by parallel copies of a, b, and c with a coloring of the same type as the majority among the three on v_1, v_2 and v_3.*

Proof of Theorem 12. Let us take a 3-CNF formula on variables $x_1, x_2, \ldots, x_n$. First, let $V := \{v_0, \ldots, v_n\}$ and take a digraph D on $V + s$ whose arc set consists of only root arcs sv_i $(i = 0, \ldots, n)$, three copy of each. Take a base $\{a, b, c\}$ of the Fano matroid and define $\mathcal{M}$ such that, for each $i \in \{0, \ldots, n\}$, the three arc sv_i form a parallel copy $\{a_i, b_i, c_i\}$ of $\{a, b, c\}$. Next use operation $COC_{b_{i-1}}(v_{i-1}, v_i)$ for $i = 1, \ldots, n$. This ensures that in every feasible packing the parallel copies of b got by $v_0, \ldots, v_n$ are colored by the same color, say, blue.

Add $v_1', \ldots, v_n'$ by $v_i' = CC_{a^i,c^i}(v_i)$ for $i = 1, \ldots, n$. Then, in every feasible packing, v_i' gets the colored base (a_i', b_i', c_i') with the same coloring as (c_i, b_i, a_i) for $i = 1, \ldots, n$. In the following construction, v_i will represent the variable x_i and v_i' its negate $\bar{x}_i$ for $i = 1, \ldots, n$. Moreover, v_0 will represent 1.

For each clause ψ of the formula, we first add w_1^ψ, w_2^ψ, w_3^ψ and w_4^ψ using operation MAJ so that it represents ψ according to (4). Finally, to ensure the truth of each clause ψ, we further use operation $AF_{b_0}(v_0, w_4^\psi)$. We claim that the formula is satisfiable if and only if $(D, \mathcal{M})$ admits a feasible packing. See Fig. 1.

Suppose that the formula has a true assignment. Then, we first construct a feasible packing restricted on $\{s, v_0, v_1, \ldots, v_n\}$ such that v_0 gets the base (a_0, b_0, c_0) colored by (red, blue, black) and each v_i $(1 \leq i \leq n)$ gets the base (a_i, b_i, c_i) colored by (red, blue, black) if $x_i = 1$ and by (black, blue, red) if $x_i = 0$. By $CC_{a^i,c^i}(v_i)$, this packing always extends on $\{v_1', \ldots, v_n'\}$ such that each v_i' gets a base formed by parallel copies of a, b, and c colored by black, blue, and red, resp., if $x_i = 1$ and by red, blue, and black, resp., if $x_i = 0$. Since the assignment satisfies the formula, by the properties of MAJ and $AF_{b_0}(v_0, w_4^\psi)$, the packing is extendable to a feasible packing on the whole vertex set of D.

Conversely, if $(D, \mathcal{M})$ has a feasible packing, then by $COC_{b_{i-1}}(v_{i-1}, v_i)$, b_i has the same color for all v_i. We set x_i in such a way that $x_i = 1$ if and only if the coloring of (a_i, b_i, c_i) is equal to that of (a_0, b_0, c_0). By $CC_{a^i,c^i}(v_i)$, each b'_i has the same color as that of b_i and the coloring of (a'_i, c'_i) is different from that of (a_i, c_i). Moreover, since $AF_{b_0}(v_0, w_4^{\psi})$ is used for each clause ψ, the base on w_4^{ψ} has the same coloring scheme as that of $\{a_0, b_0, c_0\}$ on v_0 by the property of $AF_{b_0}(v_0, w_4^{\psi})$. Thus by the property of MAJ the formula is satisfied. □

4 Concluding remarks

To get an undirected counterpart of our positive results, i.e. a characterization of the existence of a "matroid-based packing of spanning rooted-trees" for rank-2, graphic or transversal matroids, one can use [3, Corollary 1.1] and the proof after that. This extends a result of Katoh and Tanigawa [14] on these fundamental matroid classes. Moreover, with the techniques of [5], we also have extensions of these results for dypergraphs (i.e. oriented hypergraphs), hypergraphs and mixed hypergraphs. On the other hand, Problem 2 is NP-complete for dypergraphs as it is NP-complete for digraphs. Also, the proof of the NP-completeness can be applied even for the undirected case as in the construction of the NP-completeness we only add vertices with in-degree 3 one by one, and hence the ordering of the vertex addition prescribes the orientation of each edge in a rooted-tree packing.

Acknowledgments. This research was supported by the Project RIME of the laboratory G-SCOP. The authors also would like to acknowledge the support by the Hausdorff Trimester Program on Combinatorial Optimization of the Hausdorff Research Institute, the University of Bonn, where this work was partially carried out. The second author was supported by the Hungarian Scientific Research Fund – OTKA, K109240, and by the MTA-ELTE Egerváry Research Group. The fourth author was supported by JSPS Postdoctoral Fellowships for Research Abroad and JSPS Grant-in-Aid for Scientific Research (B) 25280004.

References

[1] K. Bérczi, A. Frank, Packing arborescences, in: S. Iwata, (ed.), RIMS Kokyuroku Bessatsu B23: *Combinatorial Optimization and Discrete Algorithms*, 1–31 (2010)

[2] K. Bérczi, T. Király, Y. Kobayashi, Covering intersecting bi-set families under matroid constraints, *SIAM J. Discrete Math.*, 30-3, 1758–1774 (2016)

[3] O. Durand de Gevigney, V.H. Nguyen, Z. Szigeti, Matroid-based packing of arborescences, *SIAM J. Discrete Math.*, 27, 567–574 (2013)

[4] J. Edmonds, Edge-disjoint branchings, in: B. Rustin (ed.) *Combinatorial Algorithms*, Academic Press, New York, 91–96 (1973)

[5] Q. Fortier, Cs. Király, M. Léonard, Z. Szigeti, A. Talon, Old and new results on packing arborescences, *EGRES Technical Report* No. TR-2016-04 (2016)

[6] Q. Fortier, Cs. Király, Z. Szigeti, S. Tanigawa, On packing spanning arborescences with matroid constraint, *EGRES Technical Report* No. TR-2016-18 (2016)

[7] A. Frank, On disjoint trees and arborescences, In: *Algebraic Methods in Graph Theory*, 25, Colloquia Mathematica Soc. J. Bolyai, Norh-Holland, 59–169 (1978)

[8] A. Frank, On the orientation of graphs, *J. Comb. Theory, Ser. B*, 28(3), 251–261 (1980)

[9] A. Frank, Personal communication (2013)

[10] A. Frank, T. Király, Z. Király, On the orientation of graphs and hypergraphs, *Discrete Applied Mathematics*, 131(2), 385–400 (2003)

[11] A- Frank, T. Király, M. Kriesell, On decomposing a hypergraph into k connected sub-hypergraphs, *Discrete Applied Mathematics*, 131(2), 373–383 (2003)

[12] S. Fujishige, A note on disjoint arborescences, *Combinatorica*, 30(2), 247–252 (2010)

[13] N. Kamiyama, N. Katoh, A. Takizawa, Arc-disjoint in-trees in directed graphs, *Combinatorica*, 29, 197–214 (2009)

[14] N. Katoh, S. Tanigawa, Rooted-tree decomposition with matroid constrains and the infinitesimal rigidity of frameworks with boundaries, *SIAM J. Discrete Math.*, 27, 155–185 (2013)

[15] Cs. Király, On maximal independent arborescence packing, *SIAM J. Discrete Math.*, 30-4, 2107–2114 (2016)

[16] Cs. Király, Z. Szigeti, Reachability-based matroid-restricted packing of arborescences, *EGRES Technical Report* No. TR-2016-19, `www.cs.elte.hu/egres` (2016)

[17] C.St.J.A. Nash-Williams, Edge-disjoint spanning trees of finite graphs, *J. London Math. Soc.*, 36, 445–450 (1961)

[18] J.G. Oxley, *Matroid theory*, 2nd edition, Oxford University Press (2011)

[19] L. Szegő, On covering intersecting set-systems by digraphs, *Discrete Math.*, 234, 187–189 (2001)

[20] W.T. Tutte, On the problem of decomposing a graph into n connected factors, *J. London Math. Soc., 36, 221–230 (1961)*

Embedding logical functions into the Chimera graph

Katalin Friedl[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
H-1111 Budapest, Műegyetem rakpart 3., Hungary
friedl@cs.bme.hu

László Kabódi[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
H-1111 Budapest, Műegyetem rakpart 3., Hungary
kabodil@cs.bme.com

Abstract: The Chimera graph is used in the D-Wave quantum annealer machines. Although there is a debate whether these machines are truly quantum, it is still meaningful to investigate the corresponding computational model. In this paper we show a method to embed some logical functions into the Chimera graph which can be used to solve the SAT problem using a quantum annealer.

Keywords: adiabatic quantum computation, Chimera graph, D-Wave Systems

1 Introduction

Quantum computing is a promising field of algorithmic research. The best known results are the algorithms of Grover and Shor. Grover's search finds a marked element in a list of N unordered elements in only $O(\sqrt{N})$ quantum steps and Shor's algorithm finds a prime factor of a composite number in expected polynomial time.

The model of quantum computation that is used most of the time, and also in these two famous algorithms, is the circuit model when the algorithm is built up from a small set of quantum gates, similarly to the classical model that is based on a basic set of Boolean gates.

Adiabatic computing [5] is a continuous model for quantum computations. It uses a physical process to perform quantum annealing. The problem to be solved is phrased as an optimization problem. The algorithm starts in an initial state H_i that should be an easily obtained ground state of the system. During the computation the starting Hamiltonian H_i evolves adiabatically, slowly changing but staying in ground state to reach H_p. The solution is encoded in H_p. In this type of computation the time depends on the physical process, and a challenge is to define (and create) the right Hamiltonians. It was shown in [1] that this adiabatic model is equivalent to the gate model.

Currently there is only one type of commercially available computer based on this idea, produced by D-Wave Systems, although there are doubts whether these machines really perform quantum computations. However, their structure provides an interesting computational model. In this model we do not have to deal with the Hamiltonians, the main task is to embed problems into special type of graphs (Chimera graphs).

Such embeddings are given for the general or this special adiabatic model in a few papers [2, 4, 6, 7]. They show how to represent for example an input graph in this model.

In paper [3] the first few NP-complete problems of Karp are embedded into the general adiabatic setting. Here the 3SAT problem is handled by reduction from the maximum independent set problem using the general adiabatic framework.

[1]Research is supported by OTKA-108947.

Our goal in this paper is to show a direct embedding of logical functions into the Chimera graph, specifying a possible setting for the weights of the graph.

Section 2 describes the architecture (Chimera graph), the parameters and the discrete optimization problem arising in the model of D-Wave machines. Section 3 describes the general methods used in our approach. Section 4 shows how to use these to compute the OR function of n bits, the next section describes the case of AND. Section 6 sketches how to put these together to obtain an embedding of any CNF formula.

2 The programming model of the D-Wave machine

The underlying optimization problem is the quadratic unconstrained binary optimization (QUBO) problem. For this a graph is given on N nodes, its edge set is denoted by E. The nodes and edges have weights α_i and $\beta_{i,j}$, respectively. In the corresponding QUBO problem there is a $\{0,1\}$ variable z_i to each node and the goal is to find the minimum of $\sum_{i=1}^{N} \alpha_i z_i + \sum_{\{i,j\}\in E} \beta_{i,j} z_i z_j$.

The hardware in the case of D-Wave machines uses variables $y_i \in \{-1,+1\}$, but it is easy to transform from z_i to y_i and vice versa. We will mostly use $\{0,1\}$ variables, but some ideas are easier to see in $\{-1,+1\}$.

In the case of D-Wave the underlying graph is not a complete graph. This makes formulation of problems in this setting more challenging. The computer uses a Chimera graph, which is an $m \times m$ grid of complete bipartite graphs $K_{n,n}$. (An existing choice is $m = 12, n = 4$.) Figure 1 shows a Chimera graph with a 3×3 grid and $K_{4,4}$ (a 3-4-Chimera graph). Programming the machine means setting the constants α_i and $\beta_{i,j}$. The hardware then finds the minimum of the QUBO and outputs an optimal choice for z_i.

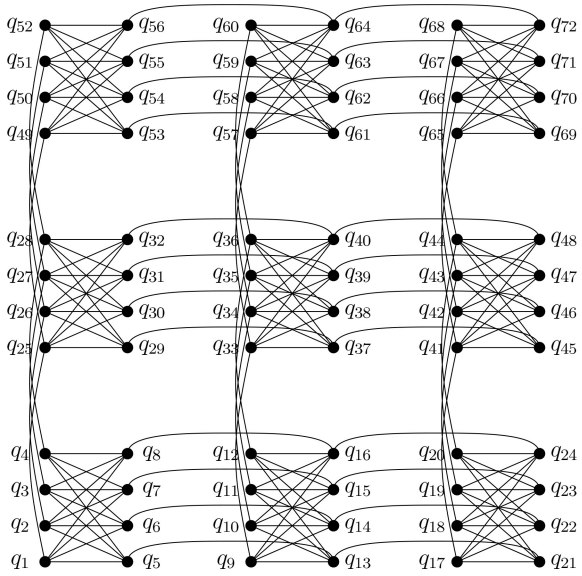


Figure 1: A 3-4-Chimera graph. Image from [8].

In a Chimera graph a node can be identified using 3 indices. The first two describe the position in the grid, the third gives its place in the corresponding bipartite graph. In the $K_{n,n}$ we number the nodes starting on the left side from top to bottom and continuing on the right side from top to bottom. For example $x_{2,3,5}$ is in the second row third column of the grid and the fifth node of the bipartite graph (that is q_{48} in Figure 1).

Using this notation the set of nodes is $\{x_{i,j,k} \mid 1 \leq i,j \leq m, 1 \leq k \leq 2n\}$. There are three kinds of edges in the graph. There are the edges of the complete bipartite graphs. The other two kinds are going between bipartite graphs. One type is the vertical connections, where $x_{i,j,k}$ is connected to $x_{i-1,j,k}$ and $x_{i+1,j,k}$ if $k \leq n$, and $2 \leq i \leq m-1$. The first and last one have only one vertical edge, $x_{1,j,k}$ is connected to $x_{2,j,k}$ and $x_{m,j,k}$ is to $x_{m-1,j,k}$. The other type is the horizontal connections, where $x_{i,j,k}$ is connected to $x_{i,j-1,k}$ and $x_{i,j+1,k}$ if $k > n$ and and $2 \leq j \leq m-1$. The first and last one have only one horizontal edge, $x_{i,1,k}$ is connected to $x_{i,2,k}$ and $x_{i,m,k}$ is to $x_{i,m-1,k}$. Notice that only the nodes on the left side of the $K_{n,n}$ have vertical edges and the nodes on the right side have horizontal ones.

In this area, embedding a graph G into a graph H means that for every vertex v of G there is a subset X_v of the vertices of H with the properties that for different vertices the sets X_v are disjoint, X_v induces a connected graph in H, and if there is an edge in G between v and w then there is an $a \in X_v$ and a $b \in X_w$ that a and b are connected be an edge in H.

One can embed a complete graph in this structure [6] which is useful to solve graph problems and also makes the definition of QUBO problems easier. But if the problem does not need a complete graph, there may be embeddings with less overhead or a cleaner design.

3 Overview of the method

The idea behind the method is to create a modular design, where one can take the appropriate modules and put them together to form arbitrary logical functions. Before we describe selected gates, we discuss the broad structure of our method.

Definition 1 *A* module *is a self-contained implementation of a small logical function.*

In our case each module is a $K_{n,n}$. Every node has a value $z_i \in \{0,1\}$.

Definition 2 *A* state *of a module is the value of its nodes.*

Definition 3 *The* value *of a state is* $\sum_{i=1}^{N} \alpha_i z_i + \sum_{\{i,j\} \in E} \beta_{i,j} z_i z_j$ *where* z_i *are the values of the nodes and* E *is the set of edges of the module.*

On the left side of the module there are three kinds of nodes: input nodes, one output node and some or none other nodes. The other nodes are not used in the module, they are only there because of the hardware. The input nodes correspond to the variables of the logical function and the output node to the value of the function.

The value of the output node is called the output of the module. The goal is to set the weights of the module such that the value of the module is minimal if and only if the output is equal to the value of the logical function.

Definition 4 *A state is* valid *if the output is the value of the function and the value of any node* $x_{i,j,k}$ *on the right side is equal to the negated value of* $x_{i,j,k-n}$ *on the left side. Otherwise the state is* invalid.

Definition 5 *We call a state* true *if it is valid and the value of the function is true. We call it* false *if it is valid and the value of the function is false.*

For an example let us examine the $\begin{pmatrix} z_1 & z_4 \\ z_2 & z_5 \\ z_3 & z_6 \end{pmatrix}$ state of a $K_{3,3}$ OR module, where z_1 and z_2 are the inputs and z_3 is the output. The state $\begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$ is true, $\begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$ is false and $\begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$ is invalid.

In our construction the weights of an edge or node depends only on its type. Let us assume that on the left side the last node is the output node, the others input nodes. We use the following parameters:

- a: the weight of all edges between $x_{i,j,k}$ and $x_{i,j,k+n}$ where $k \leq n$
- b: the weight of all edges between $x_{i,j,k}$ and $x_{i,j,\ell}$ where $k \neq \ell$, $k < n$, $\ell > n$
- c: the weight of all edges between $x_{i,j,k}$ and $x_{i,j,2n}$, where $k < n$
- d: the weight of all edges between $x_{i,j,k}$ and $x_{i,j,n}$, where $k > n$ and $k \neq 2n$
- L: the weight of all nodes $x_{i,j,k}$, where $k \leq n$
- R: the weight of all nodes $x_{i,j,k}$, where $k > n$

More precisely we work with modules that are $K_{3,3}$, where node number 1 and 2 are input nodes and 3 is the output node. For the case of larger $K_{n,n}$ the construction can be easily transformed by setting all the weights not included in the $K_{3,3}$ to be 0.

The optimization problem searches the minimum state of the QUBO problem. Our goal is to set the value of previous parameters such that all true states have the same W_t value, all false states have the same W_f value and the value of any invalid state is at least W_i. Also we want $W_t = W_f - 1$ and $W_t \leq W_i - k$, to have a $k \geq 2$ gap between the true states and the invalid states.

There might be technical constraints on the values of the parameters, but we will disregard them.

4 Embedding a logical OR

First, let us describe a module for an OR of two logical variables. It in not difficult to check that the following parameters satisfy the constraints using $k = 2$: $a = 10, b = -2, c = 6, d = \frac{2}{3}, L = -3, R = -\frac{8}{3}$. Using these parameters with a $K_{3,3}$ the value of the true states are -9, the false state is -8 and the invalid states are at least -7.

Attaching two modules together is a simple additive step. To obtain an OR function with 3 variables we use two neighbouring modules of the grid. The first represents $r_1 = p_1 \vee p_2$ and the second $r = r_1 \vee p_3$.

In order to do this one has to be able to copy the value of a node to another node. Because the QUBO is a simple sum of the different products, one can simply set the value of a few external edges, without modifying the inside of a module.

To copy the result to a new node, we simply use a sufficiently large positive or negative edge, and compensate its effect on the connected nodes. It is easier to see how this works with variables $y_i \in \{-1, +1\}$. In this case to force two connected nodes to be the same, we must use a negative edge, otherwise a positive one. For the $\{0, 1\}$ case, we first transform the variables to $\{-1, +1\}$, then use the appropriate edge and we get the weights needed to the $\{0, 1\}$ case. By this method the edge weight $w_{i,j}$ of the $\{-1, +1\}$ case transforms to the case $z_i, z_j \in \{0, 1\}$ as follows: $(2z_i-1)(2z_j-1)w_{i,j} = 4z_iz_jw_{i,j}-2(z_i+z_j)w_{i,j}+w_{i,j}$. The last term does not depend on the variables z_i, so it is not important from the point of view of minimalization. The others mean the we need to add $-\frac{1}{2}$ times the weight of the edge to the weight of nodes it connects. As before, $w_{i,j}$ is negative when copying and negative for negation.

Because we set the value of the false state to be one more than the value of the true states, we must compensate for it, so we add one to the weight of the edge that copies the result of the first module to the second one. This ensures that the minimum states include the ones, where there are some false modules, but the overall value of the function is true.

5 The logical AND

The logical AND function can be obtained from the logical OR and negations. But we think the design is cleaner if we make a separate AND module.

Applying the same constraints to the logical AND function, we obtain the following parameters: $a = 10, b = -\frac{2}{3}, c = 1, d = 5, L = -3, R = -\frac{8}{3}$. We deliberately chose a, L and R the same as in the case

of OR. Using these parameters with a $K_{3,3}$ the value of the true states are -9, the false state is -8 as before, and the invalid states are at least -6.6667.

Attaching the modules together is almost the same as in the OR case. The main difference is that in the AND function the result is only true if all the variables are true, so we do not need to add one to the weight of the edge that copies the result. (We can, the results will be the same, but we do not need to.)

6 The SAT problem

Embedding a general SAT problem using the previous modules is easy if the grid is large enough. The logical function has to be in CNF form. Then each clause gets its own column in the grid.

Each variable of the formula has its own row. For a logical variable p_i all $x_{i,j,1}$ correspond to p_i and $x_{i,j,n+1}$ to $\neg p_i$. If the clause in the jth column needs the negated version of the variable, then instead of copying the value (by negative edge weight) to that column we use positive edge weight to obtain the negation of the variable.

To implement the whole CNF formula there are three kinds of modules. The OR module, the AND module, and a copy module. The copy module keeps the value of one of its inputs, and copies the other to its output.

During this process, because the kth node of one bipartite graph is only connected to the kth node of the neighbouring bipartite graphs, the input and the output must switch places alternately. With the help of this, in a column we can move the partial results to the literals included in that clause where the OR module can be applied.

The results of the OR modules are copied into one row, in which we use AND modules. One of the input nodes of these AND modules, coming from the OR modules are always at the same position. The other input node that corresponds to the the result of the previous AND and the output node switch places alternately as we move from one AND module to the next.

7 Conclusion

In this paper we proposed a framework for constructing modules from small logical functions and applied it to construct OR and AND modules. From these, we made an embedding for any CNF into the Chimera graph. For a CNF containing n variables and m clauses, we need a Chimera graph with $n+1$ rows and m columns, so a $\max(n+1, m)$-3-Chimera graph. This embedding is not optimal, the number of nodes can be reduced, but our goal was not to find the optimal embedding, rather a clean and simple one. Later research should be done to optimize these results.

References

[1] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. LLoyd, O. Regev, Adiabatic quantum computation is equivalent to standard quantum computation, *SIAM J. Computing* **Vol. 37, pp 166-194** (2007)

[2] V. Choi, Minor embedding in adiabatic quantum copmutation: 1. The parameter setting problem *Quantum Information Processing* **Vol.7, pp. 193-209** (2008)

[3] V. Choi, Adiabatic quantum algorithms for the NP-complete maximum weight independent set, exact cover and 3SAT problems *arXiv:quant-ph/1004.2226* (2010)

[4] C.S. Claude, E. Claude, M.J. Dinneen, Guest column: Adiabatic quantum computing challenges *ACM SIGACT News* **Vol.45, pp 40-61** (2015)

[5] E. Fahri, J. Goldstone, S. Gutmann, M. Sipser, Quantum computation by adiabatic evolution, *arXiv:quant-ph/0001106* (2000)

[6] C. Klymko, B. D. Sullivan, T. S. Humble, Adiabatic quantum programming: Minor embedding with hard faults *Quantum Information Processing* **Vol.13, pp 709-729** (2014)

[7] E. G. Rieffel, D. Venturelli, B. OGorman, M. B. Do, E. M. Prystay, V. N. Smelyanskiy, Vadim N, A case study in programming a quantum annealer for hard operational planning problems *Quantum Information Processing* **Vol.14, pp 1-36** (2015)

[8] V. N. Smelyanskiy, E. G. Rieffel, S. I. Knysh, C. P. Williams, M. W. Johnson, M. C. Thom, W. G. Macready, K. L. Pudenz, A near-term quantum computing approach for hard computational problems in space exploration. *arXiv:quant-ph/1204.2821* (2012)

The Random Assignment Problem with Submodular Constraints on Goods

Satoru FUJISHIGE[1]

Research Institute for Mathematical Sciences
Kyoto University
Kyoto 606-8502, Japan
fujishig@kurims.kyoto-u.ac.jp

Yoshio SANO[2]

Division of Information Engineering
Faculty of Engineering, Information and Systems
University of Tsukuba
Ibaraki 305-8573, Japan
sano@cs.tsukuba.ac.jp

Ping ZHAN

Department of Communication and Business
Edogawa University
Nagareyama, Chiba 270-0198, Japan
zhan@edogawa-u.ac.jp

Abstract: Problems of allocating indivisible goods to agents in an efficient and fair manner without money have long been investigated in the literature. The random assignment problem is one of them, where we are given a fixed feasible (available) set of indivisible goods and a profile of ordinal preferences over the goods, one for each agent. Then, using lotteries, we determine an assignment of goods to agents in a randomized way. A seminal paper of Bogomolnaia and Moulin (2001) shows a probabilistic serial (PS) mechanism to give an efficient and envy-free solution to the assignment problem.

In this paper we consider an extension of the random assignment problem to that with submodular constraints on goods. It is revealed that the approach of the PS mechanism by Bogomolnaia and Moulin is powerful enough to solve the random assignment problem with submodular (matroidal and polymatroidal) constraints. Under the agents' ordinal preferences over goods we show the following.

1. The obtained PS solution for the problem with unit demands and matroidal constraints is ordinally efficient, envy-free, and strategy-proof with respect to the associated stochastic dominance relation.
2. For the multi-unit demand and polymatroidal constraint problem the PS solution is ordinally efficient and envy-free but is not strategy-proof in general. However, we show that under a mild condition (that is likely to be satisfied in practice) the PS solution is a weakly Nash equilibrium.

Keywords: Random assignment, probabilistic serial mechanism, ordinal preference, polymatroids, submodular optimization, Nash equilibrium

1 Introduction

Problems of allocating indivisible goods to agents in a fair and efficient manner without money have long been investigated in the literature (see, e.g., [23, 25, 1, 5, 18, 19, 4, 15, 16, 3, 24]). Suppose that

[1]Research is supported by JSPS KAKENHI Grant Number JP25280004.
[2]Research is supported by JSPS KAKENHI Grant Numbers JP15K20885, JP16H03118.

we are given a fixed feasible (available) set of indivisible goods and a profile of ordinal preferences over the goods, one for each agent. Then, using lotteries, we determine an assignment of goods to agents in a randomized way. A seminal paper of Bogomolnaia and Moulin [5] shows a probabilistic serial (PS) mechanism to give an efficient and envy-free solution to the assignment problem.

In this paper we consider an extension of the random assignment problem to that with submodular constraints on goods in two cases:

1. Agents have unit demands and the family of feasible sets of goods forms a family of bases of a matroid. (The original problem in [5] is concerned with a matroid having only one base.)

2. Agents have multi-unit demands and the set of feasible integral vectors of goods forms an integral polymatroid. (A polymatroid having only one base is treated in [6, 16, 19].)

It is revealed that the approach of the PS mechanism by Bogomolnaia and Moulin [5] is powerful enough to solve the random assignment problem with submodular (matroidal and polymatroidal) constraints. Under the agents' ordinal preferences over goods we show the following.

1. The obtained PS solution for the problem with unit demands and matroidal constraints is ordinally efficient, envy-free, and strategy-proof with respect to the partial order defined by the stochastic dominance relation introduced by Bogomolnaia and Moulin [5].

2. For the multi-unit demand and polymatroidal constraint problem the PS solution is ordinally efficient and envy-free but is not strategy-proof in general. However, we show that under a mild condition (that is likely to be satisfied in practice) the PS solution is a weakly Nash equilibrium.

The well-known Birkhoff-von Neumann theorem on bi-stochastic matrices shows that every bi-stochastic matrix is expressed as a convex combination of permutation matrices, which plays a crucial rôle in designing the probabilistic serial mechanism developed by Bogomolnaia and Moulin [5]. On the other hand, our extended probabilistic serial mechanism heavily depends on the results of submodular optimization such as the integrality of the independent flow polyhedra ([9, 11]), which generalizes the Birkhoff-von Neumann theorem.

The present paper is organized as follows. Section 2 gives some definitions and preliminaries to be used later. In Section 3 we precisely describe the random assignment problem with submodular (polymatroidal and matroidal) constraints. In Section 4 we show a procedure to find a solution in the convex hull of the feasible allocations (as an expected allocation) in an efficient and fair manner. In Section 5 we examine the issue of strategy-proofness of our solution mechanism. Section 6 shows how to design a lottery efficiently to get the desired expected allocation given in Section 4. Section 7 concludes this paper.

The present paper is based on the authors' working papers [12] and [13].

2 Definitions and Preliminaries

In this section we give definitions of some concepts from the theory of matroids and polymatroids and also give preliminary lemmas and theorems to be used in the following (see, e.g., [11]).

Let E be a nonempty finite set. For any subset $X \subseteq E$ denote by χ_X the characteristic vector of X in $\mathbb{R}^E$, i.e., $\chi_X(e) = 1$ for $e \in X$ and $\chi_X(e) = 0$ for $e \in E \setminus X$. We also write χ_e instead of $\chi_{\{e\}}$ for $e \in E$.

A pair (E, ρ) of set E and a function $\rho : 2^E \to \mathbb{R}_{\geq 0}$ is called a *polymatroid* if the following three conditions hold (see, e.g., [7, 11]).

1. $\rho(\emptyset) = 0$.

2. For any $X, Y \in 2^E$ with $X \subseteq Y$ we have $\rho(X) \leq \rho(Y)$.

3. For any $X, Y \in 2^E$ we have $\rho(X) + \rho(Y) \geq \rho(X \cup Y) + \rho(X \cap Y)$.[1]

[1] A set function satisfying these inequalities is called a *submodular function* and the negative of a submodular function is called a *supermodular function*. A function that is submodular and at the same time supermodular is called a *modular function*.

The function ρ is called the *rank function* of the polymatroid (E, ρ). We assume $\rho(E) > 0$ in the sequel.

For a given polymatroid (E, ρ), let $\mathrm{B}(\rho)(\subseteq \mathbb{R}^E)$ be the *base polytope* of the polymatroid (see, e.g., [11]), which is given by

$$\mathrm{B}(\rho) = \{x \in \mathbb{R}^E \mid \forall X \subset E : x(X) \leq \rho(X),\ x(E) = \rho(E)\}, \tag{1}$$

where for any $X \subseteq E$ we define $x(X) = \sum_{e \in X} x(e)$. It should be noted that $\mathrm{B}(\rho) \subseteq \mathbb{R}^E_{\geq 0}$. Also consider the lower hereditary closure of the base polytope $\mathrm{B}(\rho)$ given by

$$\mathrm{P}(\rho) = \{x \in \mathbb{R}^E \mid \forall X \subseteq E : x(X) \leq \rho(X)\}, \tag{2}$$

which is called the *submodular polyhedron* associated with ρ. The polytope $\mathrm{P}_{(+)}(\rho) \equiv \mathrm{P}(\rho) \cap \mathbb{R}^E_{\geq 0}$ is called the *independence polytope* of polymatroid (E, ρ) and each vector in $\mathrm{P}_{(+)}(\rho)$ is called an *independent vector*. Given a vector $x \in \mathrm{P}(\rho)$, a subset X of E is called *tight* for x (or *x-tight* for short) if we have $x(X) = \rho(X)$, and there exists a unique maximal x-tight set, denoted by $\mathrm{sat}(x)$, which is equal to the union of all tight sets for x. We also have

$$\mathrm{sat}(x) = \{e \in E \mid \forall \alpha > 0 : x + \alpha\chi_e \notin \mathrm{P}(\rho)\}, \tag{3}$$

which is the set of elements $e \in E$ for which we cannot increase $x(e)$ without leaving $\mathrm{P}(\rho)$. Moreover, for $x \in \mathrm{P}(\rho)$ and $e \in \mathrm{sat}(x)$ define

$$\mathrm{dep}(x, e) = \{e' \in E \mid \exists \alpha > 0 : x + \alpha(\chi_e - \chi_{e'}) \in \mathrm{P}(\rho)\}. \tag{4}$$

which is the unique minimal x-tight set containing e.

For any polymatroid (E, ρ) with an integer-valued rank function ρ define

$$\mathrm{B}_{\mathbb{Z}}(\rho) = \mathrm{B}(\rho) \cap \mathbb{Z}^E, \qquad \mathrm{P}_{\mathbb{Z}}(\rho) = \mathrm{P}(\rho) \cap \mathbb{Z}^E. \tag{5}$$

The following is well known (see, e.g., [11]).

Theorem 2.1 *When (E, ρ) is a polymatroid with an integer-valued rank function ρ, $\mathrm{B}(\rho)$ (resp. $\mathrm{P}(\rho)$) is the convex hull of $\mathrm{B}_{\mathbb{Z}}(\rho)$ (resp. $\mathrm{P}_{\mathbb{Z}}(\rho)$). Moreover, when (E, ρ) is a matroid, $\mathrm{B}_{\mathbb{Z}}(\rho)$ (or $\mathrm{P}_{(+)}(\rho) \cap \mathbb{Z}^E$) is exactly the set of all the characteristic vectors of bases (or independent sets) of matroid (E, ρ).*

Consider a capacitated network $\mathcal{N} = (G = (V, A), S^+, S^-, c, (S^+, \rho^+), (S^-, \rho^-))$ with polymatroids on sets $S^+, S^- \subset V$. Here G is the underlying graph with vertex set V and arc set A, and S^+ and S^- are disjoint subsets of V and are, respectively, the set of sources and that of sinks. Furthermore, we have a capacity function $c : A \to \mathbb{R}_{\geq 0}$ and a pair of polymatroids (S^+, ρ^+) and (S^-, ρ^-). A function $\varphi : A \to \mathbb{R}$ is called an *independent flow* in $\mathcal{N}$ if it satisfies

$$0 \leq \varphi(a) \leq c(a) \qquad (\forall a \in A), \tag{6}$$

$$\partial\varphi(v) = 0 \qquad (\forall v \in V \setminus (S^+ \cup S^-)), \tag{7}$$

$$\partial^+\varphi \in \mathrm{P}_{(+)}(\rho^+), \qquad \partial^-\varphi \in \mathrm{P}_{(+)}(\rho^-), \tag{8}$$

where $\partial\varphi(v) = \sum_{(v,w) \in A} \varphi(v, w) - \sum_{(w,v) \in A} \varphi(w, v)$ for all $v \in V$ and $\partial^\pm\varphi : S^\pm \to \mathbb{R}$ are defined by $\partial^+\varphi(v) = \partial\varphi(v)$ for all $v \in S^+$ and $\partial^-\varphi(v) = -\partial\varphi(v)$ for all $v \in S^-$. We may consider a cost function $\gamma : A \to \mathbb{R}$, which gives a problem of finding a minimum-cost independent flow in $\mathcal{N}$. This is called the *independent flow* problem [9] and is equivalent to what is called the *submodular flow* problem (see [11]).

We have the following integrality theorem ([9, 11]), which plays a crucial rôle in validating our approach based on the PS mechanism of Bogomolnaia and Moulin [5].

Theorem 2.2 *Let $P^* \subset \mathbb{R}^A$ be the set of all independent flows in network $\mathcal{N} = (G = (V, A), S^+, S^-, c, (S^+, \rho^+), (S^-, \rho^-))$. If c and $\rho^\pm$ are integer-valued, then P^* is an integral polytope, i.e., P^* is a convex polytope such that every extreme point of P^* is an integral vector.*

3 Description of the Random Assignment Problem

We give a precise definition of the random assignment problem with polymatroidal constraints and later examine the problem with matroidal constraints as a special case.

Let $N = \{1, 2, \cdots, n\}$ be a set of agents and E be a set of goods. Each good $e \in E$ should be considered as a type of good and the number of available good e can be more than one. Each agent $i \in N$ wants to obtain a certain amount of goods, denoted by $d(i) \in \mathbb{Z}_{>0}$, in total. We refer to $d(i)$ as the *demand upper bound* of agent i. The vector $d = (d(i) \mid i \in N) \in \mathbb{Z}_{>0}^N$ is called the *demand vector*. For each $i \in N$ and $e \in E$ let $x^i(e)$ be the number of copies of good e that agent i obtains. Then we must have

$$x^i(E) \equiv \sum_{e \in E} x^i(e) \leq d(i) \tag{9}$$

for every agent $i \in N$. Let $\mathbf{B} \subseteq \mathbb{Z}_{\geq 0}^E$ be the set of all available vectors of goods in the market that is given by $\mathbf{B} = \mathrm{B}_{\mathbb{Z}}(\rho)$ for a polymatroid (E, ρ) with an integer-valued rank function ρ. Since the sum of vectors $\sum_{i \in N} x^i$ must be available in the market, we have the following constraint.

$$\sum_{i \in N} x^i \in \mathrm{B}_{\mathbb{Z}}(\rho). \tag{10}$$

We assume that $\rho(E) \leq d(N)$.

Define $\mathbf{A} \subseteq \mathbb{Z}_{\geq 0}^{N \times E}$ to be the set of all functions $\varphi : N \times E \to \mathbb{Z}_{\geq 0}$ such that vectors given by $x^i = (\varphi(i, e) \mid e \in E)$ for all $i \in N$ satisfy (9) and (10). Every $\varphi \in \mathbf{A}$ determines a feasible allocation $x^i = (\varphi(i, e) \mid e \in E)$ for each agent $i \in N$.

Consider an independent-flow network $\mathcal{N} = (G = (S^+, S^-; A), c, (S^+, \rho^+), (S^-, \rho^-))$, where $S^+ = N$, $S^- = E$, $G = (S^+, S^-; A)$ is a complete bipartite graph with vertex bi-partition (S^+, S^-) and arc set $A = S^+ \times S^-$, $c(a) = +\infty$ (a sufficiently large positive integer), (S^-, ρ^-) is an integral polymatroid with rank function $\rho^- = \rho$ appearing in (10), and (S^+, ρ^+) is a polymatroid with a rank function ρ^+ given by $\rho^+(X) = \min\{d(X), \rho(E)\}$ $(X \subseteq S^+ = N)$. For simplicity we also denote the present independent-flow network by $\mathcal{N} = (N, E, c, d, (E, \rho))$.

Then from Theorem 2.1 we can easily see the following.

Lemma 3.1 *The set* $\mathbf{A}$ *is exactly the set of integer-valued independent flows* $\varphi : S^+ \times S^- \to \mathbb{Z}_{\geq 0}$ *in network* $\mathcal{N} = (N, E, c, d, (E, \rho))$.

(See Figure 1.)

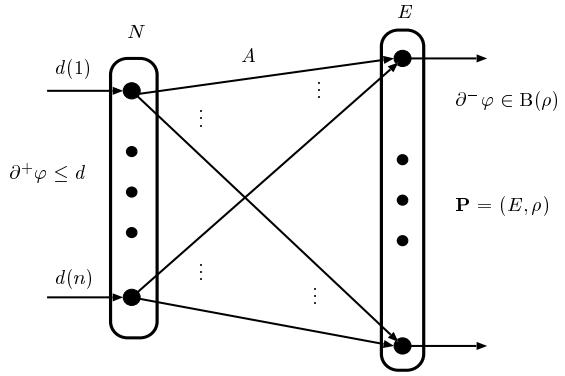


Figure 1: An independent-flow network $\mathcal{N}$.

Because of Theorem 2.2 and Lemma 3.1 we also have the following.

Corollary 3.2 *The set of all (real-valued) independent flows φ in $\mathcal{N} = (N, E, c, d, (E, \rho))$ is the convex hull* $\mathrm{Conv}(\mathbf{A})$ *of all integer-valued independent flows in $\mathcal{N}$.*

We suppose that each agent $i \in N$ has an ordinal *preference* $\succ_i$ over set E of goods, which is a linear ordering of E. Let agent i's preference be given by

$$L^i : \ e^i_1 \succ_i e^i_2 \succ_i \cdots \succ_i e^i_m, \tag{11}$$

where $\{e^i_1, e^i_2, \cdots, e^i_m\} = E$ and e^i_1 is the most favorite good for agent i. Let $\mathcal{L}$ be the profile of preferences L^i $(i \in N)$.

Since we must make a decision on how to allocate goods in a fair manner without money, we may consider a lottery, which is represented by a probability distribution p over $\mathbf{A}$, i.e., $p : \mathbf{A} \to \mathbb{R}_{\geq 0}$ satisfying $\sum_{\varphi \in \mathbf{A}} p(\varphi) = 1$. Then the *expected* allocation of goods is given by

$$\mathbf{E}\{\varphi\} = \sum_{\varphi \in \mathbf{A}} p(\varphi)\varphi, \tag{12}$$

where precisely speaking, the left-hand side is the expectation of a random variable φ with its probability distribution p on $\mathbf{A}$ while φ appearing in the right-hand side is a variable taking on values of $\mathbf{A}$. It should be noted that the set of all expected allocations $\mathbf{E}\{\varphi\}$ of (12) for all possible probability distributions p is exactly the convex hull $\mathrm{Conv}(\mathbf{A})$ of $\mathbf{A}$ and that every lottery picks up a point from among $\mathrm{Conv}(\mathbf{A})$. Moreover, when designing a lottery, it is crucial to see that given any (desired) expected allocation $\mathbf{E}\{\varphi\}$ in $\mathrm{Conv}(\mathbf{A})$, in order to realize a lottery that gives the expected allocation $\mathbf{E}\{\varphi\}$ we need at most $|N| \times |E| + 1$ (extreme) points in $\mathbf{A}$ that has positive probabilities of occurrence because of Carathéodory's theorem on convex polytopes.

Consequently, our problem becomes the following two:

1. Find a point $\bar{\varphi} (= \mathbf{E}\{\varphi\})$ from among the polytope $\mathrm{Conv}(\mathbf{A})$ in an efficient and fair manner according to the preference profile $\mathcal{L} = (L^i \mid i \in N)$. (Precise definitions of efficiency and fairness will be given later.)

2. Construct a lottery by finding a representation of $\bar{\varphi}$ as a convex combination of integral points of polytope $\mathrm{Conv}(\mathbf{A})$. The coefficients of the convex combination provide us with positive probabilities of a probability distribution over $\mathbf{A}$ that leads us to $\bar{\varphi} = \mathbf{E}\{\varphi\}$ in (12).

It heavily depends on the structure of the set $\mathbf{A}$ of feasible allocations whether we can find a desired expected solution $\bar{\varphi}$ and construct a lottery to realize $\bar{\varphi}$ in a computationally efficient way. Fortunately, it follows from Theorem 2.2 and Corollary 3.2 that $\mathbf{A}$ is the independent flow polytope and has a nice combinatorial structure as shown in the literature (see, e.g., [11]). We will see that the probabilistic serial (PS) mechanism by Bogomolnaia and Moulin [5] works surprisingly well for these general problem settings with submodular constraints.

The problem considered here includes the following as special cases.

(a) The ordinary random assignment problem considered in the literature is mostly the case where $d = \mathbf{1} \in \mathbb{Z}^N_{>0}$ and $\mathbf{B} = \{\mathbf{1}\} \subseteq \mathbb{Z}^E_{>0}$ (e.g., [5, 18, 4]). Here $\mathbf{1}$ denotes a vector of all ones of appropriate dimension (determined by the context).

(b) Kojima [19], Aziz [2], and Heo [16] considered a multi-unit demand case where $d \in \mathbb{Z}^N_{>0}$ and $\mathbf{B} = \{b\} \subseteq \mathbb{Z}^E_{>0}$ for some $b \in \mathbb{Z}^E_{>0}$.

Note that when $\mathbf{B}$ is a singleton set as in (a) and (b) above, the underlying polymatroid (E, ρ) has the unique base and the rank function ρ is modular.

In the following we use $N \times E$ matrices P to express expected allocations $\varphi \in \mathrm{Conv}(\mathbf{A})$ by identifying φ with $P = (\varphi(i, e) \mid i \in N, \ e \in E)$, which is often employed in the literature. So we may write $P \in \mathrm{Conv}(\mathbf{A})$, for example. When φ corresponds to P, φ is sometimes written as φ_P.

An efficient and fair expected allocation will be found with respect to the *stochastic dominance relation* (*sd-dominance relation* for short) $\succeq_i^{\mathrm{d}}$ for each agent $i \in N$ on expected allocations defined as follows. For any $P, Q \in \mathrm{Conv}(\mathbf{A})$, putting $P_i = (P(i,e) \mid e \in E)$ and $Q_i = (Q(i,e) \mid e \in E)$ for all $i \in N$,

$$P_i \succeq_i^{\mathrm{d}} Q_i \iff \forall \ell \in \{1, \cdots, m\} : \sum_{k=1}^{\ell} P(i, e_k^i) \geq \sum_{k=1}^{\ell} Q(i, e_k^i). \tag{13}$$

We say an expected allocation P is *sd-dominated* by Q if we have $Q_i \succeq_i^{\mathrm{d}} P_i$ for all $i \in N$ and $P \neq Q$. We say that P is *ordinally efficient* if P is not sd-dominated by any other expected allocation in $\mathrm{Conv}(\mathbf{A})$ (cf. [5]).

Also, we say an expected allocation P is *normalized envy-free* ([16]) with respect to a profile of ordinal preferences $\succ_i$ for all $i \in N$ if for all $i, j \in N$ we have $\dfrac{1}{d(i)} P_i \succeq_i^{\mathrm{d}} \dfrac{1}{d(j)} P_j$.

4 Finding an Efficient and Fair Expected Allocation

We first show a procedure Algorithm 1 which is an extension of the PS method of Bogomolnaia and Moulin [5] and will then show that the computed point in $\mathrm{Conv}(\mathbf{A})$ is an efficient and fair expected allocation.

Let us define the base $x_P^* \in \mathrm{B}(\rho)$ associated with an allocation $P \in \mathrm{Conv}(\mathbf{A})$ by

$$x_P^* \equiv \sum_{i \in N} P_i. \tag{14}$$

Recall that for each $i \in N$ agent i's preference is given by (11), where $\{e_1^i, e_2^i, \cdots, e_m^i\} = E$ and e_1^i is the most favorite good for agent i, and $\mathcal{L}$ is the profile of preferences L^i $(i \in N)$. Based on the collection (a multiset) of the first (most favorite) elements e_1^i of all agents $i \in N$, define a nonnegative integral vector $b(\mathcal{L}) \in \mathbb{Z}_{\geq 0}^E$ by

$$b(\mathcal{L}) = \sum_{i \in N} d(i) \chi_{e_1^i}, \tag{15}$$

where note that we may have $e_1^i = e_1^j$ for distinct $i, j \in N$ and $d(i)$ is the integral demand upper bound of agent $i \in N$.

We also denote the random assignment problem by $\mathbf{RA} = (N, E, \mathcal{L} = (L^i \mid i \in N), d = (d(i) \mid i \in N), (E, \rho))$.

During the execution of the following algorithm the current preference lists L^i may get shorter because of removal of exhausted (or saturated) goods.

Algorithm 1

Input: A random assignment problem $\mathbf{RA} = (N, E, \mathcal{L}, d, (E, \rho))$.
Output: An expected allocation $P : N \times E \to \mathbb{R}_{\geq 0}$.

Step 0: For each $i \in N$ put $x^i \leftarrow \mathbf{0} \in \mathbb{R}^E$ (the zero vector), and $x^* \leftarrow \mathbf{0} \in \mathbb{R}^E$.
Put $S_0 \leftarrow \emptyset$, $p \leftarrow 1$, and $\lambda_0 \leftarrow 0$.

Step 1: For current (updated) $\mathcal{L} = (L^i \mid i \in N)$, using $b(\mathcal{L})$ in (15), compute

$$\lambda_p = \max\{t \geq \lambda_{p-1} \mid x^* + (t - \lambda_{p-1}) b(\mathcal{L}) \in \mathrm{P}(\rho)\}. \tag{16}$$

For each $i \in N$ put $x^i \leftarrow x^i + (\lambda_p - \lambda_{p-1}) d(i) \chi_{e_1^i}$.
Put $x^* \leftarrow x^* + (\lambda_p - \lambda_{p-1}) b(\mathcal{L})$ and $S_p \leftarrow \mathrm{sat}(x^*)$.

Step 2: Put $T_p \leftarrow S_p \setminus S_{p-1}$.
Update L^i $(i \in N)$ by removing all elements of T_p from current L^i $(i \in N)$.

Step 3: If $\rho(S_p) < \rho(E)$, then put $p \leftarrow p + 1$ and go to Step 1.

Otherwise ($\rho(S_p) = \rho(E)$) put $P(i,e) \leftarrow x^i(e)$ for all $i \in N$ and $e \in E$.
Return P.

As in [5], the parameter t can be considered as time and each agent $i \in N$ eats the current top good e^i_1 at the rate $d(i)$ per unit time.

To see the behavior of the procedure Algorithm 1 let us consider an illustrative example given as follows.

Example: Consider $N = \{1,2,3,4\}$ and $E = \{a,b,c,d\}$. Let (E,ρ) be a polymatroid with a rank function given by

$$\rho(X) = \begin{cases} 4|X| & \text{if } |X| \le 2 \\ 8 & \text{if } |X| > 2 \end{cases} \quad (\forall X \subseteq E). \tag{17}$$

Note that (E,ρ) here is a symmetric polymatroid. Suppose that preferences of all agents are given as follows.

$i \in N$	preference L^i
1	$a \succ_1 b \succ_1 c \succ_1 d$
2	$a \succ_2 c \succ_2 b \succ_2 d$
3	$a \succ_3 c \succ_3 d \succ_3 b$
4	$b \succ_4 a \succ_4 d \succ_4 c$

Let $d = (4,2,1,1)$ be a demand vector. Then by Algorithm 1 we have $P \in \mathbb{R}^{N\times E}_{\ge 0}$, as an $N \times E$ matrix, given as follows.

$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} \frac{16}{7} & \frac{12}{7} & 0 & 0 \\ \frac{8}{7} & 0 & \frac{6}{7} & 0 \\ \frac{4}{7} & 0 & \frac{3}{7} & 0 \\ 0 & \frac{4}{7}+\frac{3}{7} & 0 & 0 \end{pmatrix} \end{array},$$

where

$$b(\mathcal{L}) = \begin{array}{cccc} a & b & c & d \\ \big(4+2+1, & 1, & 0, & 0\big), \end{array} \quad S_1 = \{a\}, \quad \lambda_1 = \tfrac{4}{7} \ \text{ for } \ p = 1$$

and

$$b(\mathcal{L}) = (0, 4+1, 2+1, 0), \quad S_2 = \{a,b,c,d\}, \quad \lambda_2 = \lambda_1 + \tfrac{3}{7} \ \text{ for } \ p = 2$$

to get the expected allocation P given above. Also, vectors $x^*_{\lambda_p}$, which are the restriction of x^*_P on $T_p = S_p \setminus S_{p-1}$ for $p = 1,2$, are given by

$$\begin{aligned} &T_{\lambda_1} = \{a\}, \quad T_2 = \{b,c,d\}, \\ &x^*_{\lambda_1}(a) = 4, \quad x^*_{\lambda_2}(b) = \tfrac{19}{7}, \quad x^*_{\lambda_2}(c) = \tfrac{9}{7}, \quad x^*_{\lambda_2}(d) = 0. \end{aligned}$$

Hence $x^*_P = (4, \frac{19}{7}, \frac{9}{7}, 0)$. Note that $\emptyset$, $\{a\}$, $\{a,b,c\}$, and $\{a,b,c,d\}(= \mathrm{sat}(x^*_P))$ are tight sets for x^*_P. □

4.1 Ordinal efficiency

The following theorem can be shown in a very similar way as the corresponding one in [5]. However, it can be seen that the given proof heavily depends on the underlying submodularity structure, especially the one used for the arguments in [10].

Theorem 4.1 *The procedure* Algorithm 1 *computes an expected allocation in* Conv($\mathbf{A}$) *that is ordinally efficient.*

4.2 Envy-freeness

We have the following theorem on normalized envy-freeness of the extended PS mechanism. The proof is actually a direct adaptation of the one given by Bogomolnaia and Moulin [5] and Schulman and Vazirani [24] for an ordinary problem setting (also see [16]). It should be noted that by Algorithm 1 every agent $i \in N$ eats $d(i)$ units of goods per unit time.

Theorem 4.2 *The procedure* Algorithm 1 *computes an expected allocation P that is normalized envy-free.*

5 Strategy-proofness

It is known that the extension of the PS mechanism of Bogomolnaia and Moulin to the case of multi-unit demands cannot be weakly strategy-proof in general ([6, 16, 19, 3]). Therefore, our polymatroidal extension is not weakly strategy-proof in general either.

Note that a solution mechanism M is *weakly strategy-proof* if for *every* input preference profile $\mathcal{L}$ the mechanism M gives a solution (an expected allocation) P such that every misreport of every agent i's preference results in a solution Q satisfying that Q_i does not sd-dominate P_i for i. Here the strategy-proofness is concerned with the mechanism.

5.1 Weakly Nash equilibria

Let us consider the concept of a *weakly Nash equilibrium* ([8, 17]), which is a property of the obtained solution. For a *given input profile* we say that the solution P obtained by the mechanism M is called a *weakly Nash equilibrium* if every misreport of every agent i's preference results in a solution Q satisfying that Q_i does not sd-dominate P_i for i. The solution obtained by the extended PS mechanism for multi-unit demands was investigated from the point of view of the weakly Nash equilibrium in [8, 17].

We examine our polymatroidal extension and give a certain (useful) sufficient condition for our solution to be a weakly Nash equilibrium. The result, Theorem 5.1 given below, seems to be new even for the ordinary multi-unit demand case where the base polytope consists of a single base, i.e., $\mathrm{B}(\rho) = \{b\}$ for some $b \in \mathbb{Z}^E_{>0}$.

For each $e \in E$ define $N_P(e) = \{i \in N \mid P(i,e) > 0\}$.

Theorem 5.1 *Given the solution P by* Algorithm 1*, if we have $|N_P(e)| \neq 1$ for all $e \in E$, then the solution P is a weakly Nash equilibrium.*

Theorem 5.1 is rephrased as follows. (Note that matrix $P \in \mathbb{R}^{N \times E}$ has the row set N and the column set E.)

- If no column of P contains exactly one non-zero entry, the extended PS solution P computed by Algorithm 1 is a weakly Nash equilibrium.

Theorem 5.1 has very useful practical implications from the point of view of strategy-proofness. The condition that $|N_P(e)| \neq 1$ ($\forall e \in E$) is very likely to be satisfied when the number $|N|$ of 'agents' is significantly large, compared with the number $|E|$ of 'types of goods' such as the assignment of students to courses.

Related non-manipulability result was also obtained by Kojima and Manea [20], assuming the availability of utility functions. They gave a sufficient condition for their extended PS solution to be a weakly Nash equilibrium, which can be checked by the given data including utility functions. On the other hand, our condition can easily be checked by the extended PS solution computed without using any additional information about utility functions.

We also prove the weak strategy-proofness in the special case of unit demands and matroidal supplies (shown in [13]).

5.2 Weak strategy-proofness in case of unit demands and matroidal supplies

We show that when the polymatroid (E, ρ) is a matroid and agents have unit demands, the extended PS mechanism (Algorithm 1) is weakly strategy-proof, where the matroidal $\{0, 1\}$ property plays a crucial rôle.

Theorem 5.2 *When the underlying polymatroid (E, ρ) is a matroid and agents have unit demands, the extended PS mechanism given by* Algorithm 1 *is weakly strategy-proof.*

6 Designing a Lottery

Now we examine how to compute an expression of the solution P, obtained by Algorithm 1, as a convex combination of (possibly extreme) points $Q^{(k)}$ $(k \in K)$ of $\mathrm{Conv}(\mathbf{A})$ that belong to the set $\mathbf{A}$ of integral feasible allocations as follows.

$$P = \sum_{k \in K} \nu_k Q^{(k)}, \tag{18}$$

where $\nu_k > 0$ for all $k \in K$ and $\sum_{k \in K} \nu_k = 1$.

We can always compute a required convex combination representation (18) in an efficient way. With the aid of polymatroidal results achieved in [9, 10, 11, 22] we can construct a lottery to attain P by finding the expression as in (18).

Theorem 6.1 *By using* Algorithm 1 *to find the expected allocation matrix P we can generate a feasible integral allocation in strongly polynomial time whose expectation is equal to the solution P.*

7 Conclusion

We have considered the random assignment problem with submodular constraints on goods and have shown that the probabilistic serial (PS) mechanism of Bolgomolnaia and Moulin [5] can naturally be extended to give an ordinally efficient and normalized envy-free solution, while we have shown a sufficient condition (Theorem 5.1) that guarantees that the computed PS solution is a weakly Nash equilibrium, which is practically useful for problems with a large number of agents. We have also shown the weak strategy-proofness (Theorem 5.2) of the extended PS mechanism in case of unit demands and matroidal supplies.

In our earlier manuscript [12] we investigated the random assignment problem with matroidal constraints in more details, where we examined a characterization of the extended PS solution by min-cost independent flows and by lexicographic optimality, which we have omitted here.

References

[1] A. Abdulkadíroğlu and T. Sönmez: Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica* **66** (1998) 689–701.

[2] H. Aziz: Random assignment with multi-unit demands. arXiv:1401.7700v3 [cs.GT] 19 Jun 2015.

[3] A. Bogomolnaia: Random assignment: redefining the serial rule. *Journal of Economic Theory* **158** (2015) 308–318.

[4] A. Bogomolnaia and E. J. Heo: Probabilistic assignment of objects: characterizing the serial rule. *Journal of Economic Theory* **147** (2012) 2072–2082.

[5] A. Bogomolnaia and H. Moulin: A new solution to the random assignment problem. *Journal of Economic Theory* **100** (2001) 295–328.

[6] E. Budish, Y.-K. Che, F. Kojima, and P. Milgrom: Designing random allocation mechanisms: Theory and applications. *American Economic Review* **103** (2013) 585–623.

[7] J. Edmonds: Submodular functions, matroids, and certain polyhedra. *Proceedings of the Calgary International Conference on Combinatorial Structures and Their Applications* (R. Guy, H. Hanani, N. Sauer and J. Schönheim, eds., Gordon and Breach, New York, 1970), pp. 69–87.

[8] Ö. Ekici and O. Kesten: An equilibrium analysis of the probabilistic serial mechanism. *International Journal of Game theory* **45** (2016) 655–674.

[9] S. Fujishige: Algorithms for solving the independent-flow problems. *Journal of the Operations Research Society of Japan* **21** (1978) 189–204.

[10] S. Fujishige: Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research* **2** (1980) 186–196.

[11] S. Fujishige: *Submodular Functions and Optimization* Second Edition (Elsevier, 2005).

[12] S. Fujishige, Y. Sano, and P. Zhan: A solution to the random assignment problem with a matroidal family of goods. *RIMS Preprint* **RIMS-1852**, Kyoto University, May 2016.

[13] S. Fujishige, Y. Sano, and P. Zhan: An extended probabilistic serial mechanism to the random assignment problem with multi-unit demands and polymatroidal supplies. *RIMS Preprint* **RIMS-1866**, Kyoto University, November 2016.

[14] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan: A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing* **18** (1989) 30–55.

[15] T. Hashimoto, D. Hirata, O. Kesten, M. Kurino, and M. U. Ünver: Two axiomatic approaches to the probabilistic serial mechanism. *Theoretical Economics* **9** (2014) 253–277.

[16] E. J. Heo: Probabilistic assignment problem with multi-unit demands: a generalization of the serial rule and its characterization. *Journal of Mathematical Economics* **54** (2014) 40–47.

[17] E. J. Heo and V. Manjunath: Implementation in stochastic dominance Nash equilibria. *Social Choice and Welfare* **48** (2017) 5–30.

[18] A.-K. Katta and J. Sethuraman: A solution to the random assignment problem on the full preference domain. *Journal of Economic Theory* **131** (2006) 231–250.

[19] F. Kojima: Random assignment of multiple indivisible objects. *Mathematical Social Sciences* **57** (2009) 134–142.

[20] F. Kojima and M. Manea: Incentives in the probabilistic serial mechanism. *Journal of Economic Theory* **145** (2010) 106–123.

[21] N. Megiddo: Optimal flows in networks with multiple sources and sinks. *Mathematical Programming* **7** (1974) 97–107.

[22] K. Nagano: A strongly polynomial algorithm for line search in submodular polyhedra. *Discrete Optimization* **4** (2007) 349–359.

[23] A. Roth and M. Sotomayor: *Two Sided Matching: A Study in Game-Theoretic Modeling and Analysis* (Cambridge University Press, Cambridge, 1990).

[24] L. J. Schulman and V. V. Vazirani: Allocation of divisible goods under lexicographic preferences. *Proceedings of 35th IARCS Annual Conf. Foundations of Software Technology and Theoretical Computer Science* (FSTTCS 2015) (editors: Prahladh Harsha and G. Ramalingam), pp. 543–559.

[25] L. Zhou: On a conjecture by Gale about one-sided matching problems. *Journal of Economic Theory* **52** (1990) 123–135.

Rounds in a combinatorial search problem

Dániel Gerbner[1]

MTA Rényi Institute
Hungary H-1053, Budapest
Reáltanoda utca 13-15.
gerbner@renyi.hu

Máté Vizer[2]

MTA Rényi Institute
Hungary H-1053, Budapest
Reáltanoda utca 13-15.
vizermate@gmail.com

Abstract: We consider the following combinatorial search problem: we are given some excellent elements of $[n]$ and we should find at least one, asking questions of the following type: "Is there an excellent element in $A \subset [n]$?". G.O.H. Katona [6] proved sharp results for the number of questions needed to ask in the adaptive, non-adaptive and two-round versions of this problem.

We verify a conjecture of Katona by proving that in the r-round version we need to ask $rn^{1/r} + O(1)$ queries for fixed r and this is sharp.

We also prove bounds for the queries needed to ask if we want to find at least d excellent elements.

Keywords: combinatorial search, round, adaptive, non-adaptive

1 Introduction

In the most basic model of combinatorial search theory Questioner needs to find a special element x of $\{1, 2, ..., n\}(=: [n])$ by asking minimal number of questions of type "does $x \in F \subset [n]$?". Special elements are usually called defective; in this paper, following [6] we call them *excellent*. There are many generalizations of this very basic model, one can find many directions and results in the following survey papers and books: [1, 2, 3, 4, 5].

We call the *complexity* of a specific combinatorial search problem the number of the questions needed to ask by Questioner in the worst case during an optimal strategy.

For every combinatorial search problem there are at least two main approaches: whether it is *adaptive* or *non-adaptive*. In the adaptive scenario Questioner asks questions depending on the answers for the previously asked questions, however in the non-adaptive version Questioner needs to pose all the questions at the beginning.

A possible intermediate scenario is when there are r rounds for some integer $r \geq 1$ fixed at the beginning and Questioner can pose questions in the i^{th} round ($1 \leq i \leq r$) depending on the answers for the questions posed in the first $i-1$ rounds. Note that the non-adaptive version is the one-round version, and in the adaptive version there are infinitely many rounds (however it is easy to see that at most n (or some function of n) rounds are enough for most of the combinatorial search problems). There are results in the literature, when authors provide a solution for an adaptive search problem that also solves the r-round version of that problem for some r. However we could only find few examples (see e.g. [7]) where the focus of the research is how the complexity changes depending on the number of rounds. Our results fit into this line of research.

The paper is organized as follows: in Subsections 1.1 and 1.2 we define the problem and state our results and in Section 2 we prove theorem 1, finally in Section 3 we make some remarks and pose some questions.

[1]Research supported by the National Research, Development and Innovation Office NKFIH, grant K116769 and the János Bolyai Research Fellowship of the Hungarian Academy of Sciences.

[2]Research supported by the National Research, Development and Innovation Office NKFIH under the grant SNN 116095.

1.1 The model

A question of R. Chambers was answered by G.O.H. Katona [6], who determined a sharp (up to constant terms) result for the complexity of the adaptive, non-adaptive and 2-round versions of the following model.

- **Input:** $[n]$ with some (possibly zero) excellent elements.
- **Question:** is there an excellent in $A \subset [n]$?
- **Goal:** find an excellent element or state that there is none.

We denote the r-round version of this problem by $P(n, ?, 1, r)$ and denote by $|P(n, ?, 1, r)|$ its complexity. We also consider that variant of the previous model (and denote by $P(n, ?, d, r)$), when Questioner should find (at least) d excellent elements (or state that there are at most $d-1$), and also use the notation $|P(n, ?, d, r)|$ for the complexity of the latter problem.

1.2 Results

In the following theorem we verify a conjecture of Katona ([6], Conjecture 1) by determining the complexity of $P(n, ?, 1, r)$ almost exactly.

Theorem 1 *For any $r, n \geq 1$ we have:*

$$rn^{1/r} \geq |P(n, ?, 1, r)| \geq rn^{1/r} - 2r + 1.$$

We have a larger gap in case we want to find more excellent elements, whose proof will be available in the journal version.

Theorem 2 *For any $r \geq 1$ and $n \geq d \geq 2$ we have:*

$$r\lceil (d^{r-1}n)^{1/r} \rceil + d \geq |P(n, ?, d, r)| \geq r(dn)^{1/r} - 2d - r(d+1) + 2.$$

However note that for two rounds the upper and lower bounds are asymptotically equal as n tends to infinity.

Corollary 3 *For any $n \geq d \geq 2$ we have:*

$$2\lceil (dn)^{1/2} + \rceil + 2 \geq |P(n, ?, d, 2)| \geq 2\lceil (dn)^{1/2} \rceil - 4d - 2.$$

2 Proof of Theorem 1

Proof: First we prove the upper bound. To do this we describe an algorithm (given by Katona [6]). In the first round Questioner partitions $[n]$ into $\lceil n^{1/r} \rceil$ parts such that their sizes differ by at most one. Then he asks all of these parts except one, C which is one of the smaller parts. Then he picks one of the parts that were answered yes, or if there is no such part, then he picks C. In the next round he continues on the picked part recursively, i.e. he partitions it into $\lceil n^{1/r} \rceil$ parts such that their sizes differ by at most one and asks all but one of the smaller parts, and so on. In the last (the rth) round if all the answers in the previous round were no, he changes the algorithm and asks all the parts. It is easy to see that in the last round the parts are of size at most one, thus he finds an excellent element if there is any, and that in each round at most $\lceil n^{1/r} \rceil$ queries were asked.

To prove the lower bound we describe a strategy for Adversary to force Questioner to ask at least $r(n^{1/r} - \frac{r-1}{r} - 1)$ questions before reaching his goal. First we introduce the following notation. For $1 \leq i \leq r$ let $\mathcal{F}_i$ be the family of the queries asked by the Questioner in round i and $k_i := |\mathcal{F}_i|$. Let $\mathcal{F}_i^Y \subset \mathcal{F}_i$ be the family of queries that are answered yes by Adversary, and let $\mathcal{F}_i^N \subset \mathcal{F}_i$ be the family

of those queries that are answered no (and so $\mathcal{F}_i^N = \mathcal{F}_i \setminus \mathcal{F}_i^Y$). Let $G_i := \bigcup(\cup_{j=1}^{i} \mathcal{F}_j^N)$, the set of those elements that are known to be not excellent after round i. Informally we can forget about them, and restrict the underlying set to $[n] \setminus G_i$ after round i. Finally let $\mathcal{G}_i := \cup_{j=1}^{i}(\mathcal{F}_j^Y \setminus G_i)$ the set of the queries answered yes during the first i rounds restricted to $[n] \setminus G_i$, $m_i := \min\{|G| : G \in \mathcal{G}_i\}$ the cardinality of the smallest set in $\mathcal{G}_i$ and $n_i := \lfloor n_{i-1}/(k_i+1) \rfloor \geq n/\Pi_{j=1}^{i}(k_j+1) - i$ (with $n_0 = n$, and the latter inequality is an easy consequence of the fact that $k_i \geq 0$). We remark that when we describe how Adversary answers the queries in round i, we use only information that Adversary has at that point. For example, $k_1, \ldots, k_i$ are known, but k_{i+1} is not known after Questioner poses the questions in round i.

When an element appears in a question that is answered no, we know that that element cannot be excellent, thus it does not matter if a latter question contains it or not. Hence we can assume without loss of generality that no elements of G_i appear in a member of $\mathcal{F}_j$ for $j > i$.

The proof of the lower bound for the case of two rounds by Katona essentially consists of two steps. First it is shown that the first round of queries can be answered (by Adversary) in a way that either m_1 is large or all the answers are no and $|G_1|$ is relatively small. Afterwards it is shown that in the last round if $\mathcal{F}_1^Y$ is not empty, at least $m_1 - 1$ queries are needed, or if $\mathcal{F}_1^Y$ is empty, then at least $n - |G_1|$ queries are needed. Here in Lemma 4 we extend the first step to more rounds and for sake of completeness we reprove the lemma about the last step (Lemma 5).

Now we show how Adversary should answer during the first $r-1$ rounds.

Lemma 4 *Adversary can answer $\mathcal{F}_1, \ldots, \mathcal{F}_{r-1}$ such a way that for all $1 \leq t \leq r-1$ we have either:*

- *$n_t \leq m_t - 1$, or*
- *all the answers are no in the first t rounds and $|G_t| \leq n - n_t$.*

Proof: We use induction on t and let us consider round t.

If $t = 1$, then Adversary orders the elements of $\mathcal{F}_1$ in the following way:

- let $H_1 := F_1$ be one of the smallest sets in $\mathcal{F}_1$,
- for $2 \leq i \leq |\mathcal{F}_1|$ let $F_i \in \mathcal{F}_1 \setminus \{F_1, F_2, ..., F_{i-1}\}$ be such that the cardinality of $H_i := F_i \setminus \cup_{j=i}^{i-1} F_j$ is as small as possible. Note that the sets H_i are disjoint from each other.

After this if there is no i with $|H_i| \geq n_1 + 1$, then Adversary answers no for all questions in $\mathcal{F}_1$ and we clearly have $|G_1| \leq n - n/(k_1+1) \leq n - n_1$.

However if there is an i with $|H_i| \geq n_1 + 1$, then Adversary chooses the smallest such i and answers no to F_j if $j < i$ and yes if $j \geq i$. So each question in $\mathcal{F}_1^Y$ contains a least $|H_i| \geq n_1 + 1$ elements not in $\cup_{j=1}^{i-1} H_j (= G_1)$ and we are done with the case $t = 1$.

So assume that $t \geq 2$ and first consider the case when Adversary answered in the previous rounds only no answers. Then - by induction - there are at least n_{t-1} elements we do not know anything about. Adversary restricts the queries to those elements, and do the same as in the first round. That results in either that $m_t - 1 \geq n_{t-1}/(k_t+1) \geq n_t$ or only no answers and at least $n_{t-1}/(k_t+1) \geq n_t$ many elements still not appearing in any queries.

Now we assume that Adversary answered yes at least once in the first $t-1$ rounds, and then every element of $\mathcal{G}_{t-1}$ has size at least $n_{t-1} + 1$. In this case Adversary essentially does the same as in the first round, so orders the elements of $\mathcal{F}_t$ the following way (note that every element of $\mathcal{F}_t$ is in the complement of G_{t-1}):

- let $H_1 := F_1$ be one of the smallest sets in $\mathcal{F}_t$, and
- for $2 \leq i \leq |\mathcal{F}_t|$ let $F_i \in \mathcal{F}_t \setminus \{F_1, F_2, ..., F_{i-1}\}$ is such that the cardinality of $H_i := F_i \setminus \cup_{j=i}^{i-1} F_j$ is as small as possible. Note that the sets H_i are disjoint from each other.

Let us assume first that there is an i with $|H_i| \geq n_t + 1$, and consider the smallest such i. Then Adversary answers no to F_j if $j < i$ and yes if $j \geq i$. Then each question in $\mathcal{F}_t^Y$ contains a least $|H_i| \geq n_t + 1$ elements not in $\cup_{j=1}^{i-1} H_j$. This means those members of $\mathcal{G}_t$ that correspond to queries in

round t have indeed size at least $n_t + 1$. The other members - by induction - had size at least $n_{t-1} + 1$ before the round, and at most $|\cup_{j=1}^{i} H_j| \leq k_t n_t$ elements were moved to G_t, thus deleted from them in the current (t^{th}) round. Then at least $n_t + 1$ remains in each.

If there is no such i, then Adversary answers no to every question. As earlier there was a yes answer, we still have to show that $n_t \leq m_t - 1$, but this time we do not have to deal with the new queries. For the earlier queries the same argument works: at most $|\cup_{j=1}^{i} H_j| \leq k_t n_t$ elements were deleted from each set in $\mathcal{G}_{t-1}$ and we are done with the proof of Lemma 4. □

The following lemma, which deals with the last round is essentially the generalization of Lemma 3.6 in [6], however we provide a proof somewhat more compact than the one in [6], since we want to generalize the argument during the proof of Theorem 2.

Lemma 5 *For $r \geq 2$ to be able to find an excellent element in the r^{th} round Questioner needs at least $m_{r-1} - 1$ queries if there is at least one yes answer in the first $r-1$ rounds and at least $n - |G_{r-1}|$ queries are needed if all the answers were no in the first $r-1$ rounds.*

Remark 6 *Before starting the proof, note that in Lemma 5 there is no indication about Adversary's strategy during the first $r-1$ rounds. So the statement of the lemma is true for any strategy.*

PROOF: We prove lemma 5 by induction on $n - |G_{r-1}| + m_{r-1}$.

Note that if $m_{r-1} = 0$ (so there were no 'yes' answers during the first $r-1$ rounds), then we are done by the result of Katona ([6], Theorem 2.5) on the non-adaptive version of this problem.

If $m_{r-1} = 1$, then we are also done, since there is a one-element question with containing exactly one excellent element.

Using that $n - |G_{r-1}| \geq m_{r-1}$, we are done with the cases $n - |G_{r-1}| + m_{r-1} = 1, 2, 3$.

So suppose $n - |G_{r-1}| + m_{r-1} \geq 4$ and $m_{r-1} \geq 2$. We claim the following:

Claim 7 *Questioner should ask a one-element set in the r^{th} round.*

PROOF: We prove claim 7 by contradiction. Suppose all queries are of size at least two and all the answers are yes in the r^{th} round, and Questioner can point an excellent element. Let us assume all the elements in $[n] \setminus G_{r-1}$ are excellent, except the one Questioner pointed. This is compatible with the previous answers using that $m_{r-1} \geq 2$, and also with the new answers, a contradiction. □

To continue the proof of Lemma 5 we can suppose that Questioner asks a one element question ($x \in [n] \setminus G_{r-1}$) in the r^{th} round. But then Adversary can say no to $\{x\}$ first (this is compatible with the answers in the first $r-1$ rounds, since $m_{r-1} \geq 2$) and consider it as if it were asked during the first $r-1$ rounds and delete x from the remaining queries asked in the r^{th} round. Note that in this new scenario m_{r-1} can decrease by at most 1. As $|n \setminus (G_{r-1} \cup x)| < n - |G_{r-1}|$, since $x \notin G_{r-1}$ by induction we know that Questioner should ask at least $m_{r-1} - 1$ queries and we are done with the proof of Lemma 5. □

So Lemma 5 and Lemma 4 shows that we have that

$$k_1 + ... + k_{r-1} + \frac{n}{(k_1 + 1)...(k_{r-1} + 1)} - r$$

is a lower bound on $|P(n, ?, 1, r)|$. Using some reorganization and the inequality of arithmetic and geometric means we have:

$$k_1 + ... + k_{r-1} + \frac{n}{(k_1 + 1)...(k_{r-1} + 1)} - r \geq r(n^{1/r} - \frac{r-1}{r} - 1),$$

and we are done with the lower bound and with the proof of Theorem 1. □

3 Questions, Remarks

To finish this article we pose a couple questions:

• The first one is about the statement of Theorem 2. It would be interesting to find the same multiplicative factor of $n^{1/r}$ in a lower and an upper bound thus determine the asymptotic of $|P(n, ?, d, r)|$.

Note that in the case $r = \lceil \log n \rceil$ (so basically in the adaptive case) Theorem 1 does not give back the adaptive result of Katona([6]).

• It would be interesting to determine the asymptotics of $|P(n, ?, d, r)|$, when r or d is a function of n that goes to infinity with n.

• In this paper we assumed nothing in advance about the number of excellent elements. One could consider different models where we know that there are exactly, at most, or at least e excellent elements in $[n]$.

Acknowledgement

We would like to thank all participants of the Combinatorial Search Seminar at the Alfréd Rényi Institute of Mathematics for fruitful discussions, and the anonymous referee for comments.

References

[1] R. Ahlswede and I. Wegener, Search problems, John Wiley, (1987)

[2] M. Aigner, Combinatorial search, Teubner-Wiley, (1988)

[3] F. Cicalese, Fault-Tolerant Search Algorithms, Monographs in Theoretical Computer Science-*An EATCS* Series. Vol. 15. Springer-Verlag, (2013)

[4] D.-Z. Du and F. K. Hwang, Combinatorial group testing and its applications, Vol. 12. World Scientific, (1999)

[5] G.O.H. Katona, Combinatorial search problems, *A survey of combinatorial theory* 285–308. (1973)

[6] G.O.H. Katona, Finding at least one excellent element in two rounds, *Journal of Statistical Planning and Inference* **141**, 2946–2952. (2011)

[7] G. Wiener, Rounds in combinatorial search, *Algorithmica* **67(3)**,315-323. (2013)

On Ryser's Conjecture

András Gyárfás[1]

Alfréd Rényi Institute of Mathematics,
Hungarian Academy of Sciences, P.O. Box 127,
Budapest, Hungary, H-1364
gyarfas.andras@renyi.mta.hu

Zoltán Király[2]

Department of Computer Science and
MTA-ELTE Egerváry Research Group, Eötvös
Loránd University, Pázmány Péter sétány 1/C,
Budapest, Hungary, H-1117
kiraly@cs.elte.hu

Lilla Tóthmérész[2]

Department of Computer Science, Eötvös
Loránd University, Pázmány Péter sétány 1/C,
Budapest, Hungary, H-1117
tmlilla@cs.elte.hu

Abstract: A well-known special case of a conjecture attributed to Ryser (actually appeared in the thesis of Henderson [11]) states that k-uniform k-partite intersecting hypergraphs have transversals of at most $k-1$ vertices (we call this "intersecting Ryser's conjecture" or shortly IRC). An equivalent form of the conjecture in terms of coloring of complete graphs is formulated in [8]: if the edges of a complete graph K are colored with k colors, then the vertex set of K can be covered by at most $k-1$ monochromatic components. In this paper we examine some possible generalizations of IRC.

It turned out that the analogue of this conjecture for hypergraphs can be answered: Z. Király proved [12] that for $r \geq 3$ in every k-coloring of the edges of a complete r-uniform hypergraph K^r, the vertex set of K^r can be covered by at most $\lceil k/r \rceil$ monochromatic components.

We first investigate the analogue problem for complete r-uniform r-partite hypergraphs. An edge coloring of a hypergraph is called **spanning** if every vertex is incident to edges of any color used in the coloring. We propose the following analogue of IRC.

In every spanning $(r+t)$-coloring of the edges of a complete r-uniform r-partite hypergraph, the vertex set can be covered by at most $t+1$ monochromatic components.

We show that the conjecture (if true) is best possible. Our main result is that the conjecture is true for $1 \leq t \leq r-1$. We also prove a slightly weaker result for $t \geq r$, namely that $t+2$ monochromatic components are enough to cover the vertex set.

To build a bridge between complete r-uniform and complete r-uniform r-partite hypergraphs, we introduce a new notion. A hypergraph is complete r-uniform (r,ℓ)-partite if it has all r-sets that intersect each partite class in at most ℓ vertices (where $1 \leq \ell \leq r$).

Extending our results achieved for $\ell = 1$, we prove that for any $r \geq 3$, $2 \leq \ell \leq r$ and $k \geq 1 + r - \ell$, in every spanning k-coloring of the edges of a complete r-uniform (r,ℓ)-partite hypergraph, the vertex set can be covered by at most $1 + \lfloor \frac{k-r+\ell-1}{\ell} \rfloor$ monochromatic components.

[1]Research was supported in part by grant (no. K K104343) from the National Development Agency of Hungary, based on a source from the Research and Technology Innovation Fund.

[2]Research was partially supported by grant (no. K 109240) from the National Development Agency of Hungary, based on a source from the Research and Technology Innovation Fund.

We also formulate a conjecture corresponding to the t-intersecting case: If an r-uniform r-partite hypergraph H is t-intersecting (i.e., every two hyperedges meet in at least t vertices where $t < r$), then $\tau(H) \leq r - t$. In the dual language we use in the present paper, this translates to the following. Suppose a set of colors $\mathrm{Col}(e) \subseteq [k]$ (where $[k] = \{1, \ldots, k\}$) are assigned to every edge e of a complete graph K, and $|\mathrm{Col}(e)| \geq t$ for each edge. We conjecture that the vertex set of K can be covered by $k - t$ monochromatic components (if $t < k$). We prove this conjecture for the case $t > k/4$.

A 3-edge-colored K_4 shows that in IRC sometimes we need at least two components colored by the same color. Motivated by this example, we examine what fraction of the vertices can be covered by $k - 1$ monochromatic components of *different* colors in a k-edge-colored complete graph. We prove a sharp bound for this problem, moreover, we also prove that whenever it is sharp, it "comes from" a finite affine plane.

Keywords: hypergraph, monochromatic component, Ryser's conjecture

1 Introduction

For an edge-colored (hyper)graph H, let H_i denote its sub(hyper)graph consisting of edges colored by i. The connected components of H_i are called monochromatic components of color i, and a **monochromatic component** refers to a monochromatic component of color i for some i. Here connectivity is understood in its weakest sense, a hypergraph is connected if either it has only one vertex or any two distinct vertices can be connected by a sequence of edges each intersecting the next. Every hypergraph can be uniquely partitioned into connected components. Components with a single vertex are called *trivial*. If the colors used are elements of $[k]$, then we call H a k-edge-colored (hyper)graph.

Given an edge-colored hypergraph H, let $c(H)$ denote the minimum integer m such that $V = V(H)$, the vertex set of H, can be covered by m monochromatic components of H. An edge coloring of a hypergraph is called **spanning** if every vertex is incident to edges of every color used in the coloring. Note that in spanning colorings every monochromatic component is non-trivial (we suppose H is loop-free). The importance of this definition is shown in Theorem 1.

An equivalent form of IRC formulated in [8] is as follows: if K is a k-edge-colored complete graph, then $c(K) \leq k - 1$. The conjecture is true for $k \leq 5$ and seems very difficult in general (further information can be found in [4], [9]). A particular feature of the conjecture is that $c(K) \leq k$ is obvious since the k monochromatic stars centered at any given vertex cover the vertex set. Note that the conjecture is obvious for colorings that are not spanning: if a vertex is not incident to any edge in a specific color then the stars through the vertex form the required covering.

Surprisingly, the problem for hypergraphs is easier, Z. Király in [12] showed that if the edges of a complete r-uniform hypergraph K ($r \geq 3$) are colored with k colors, then $c(K) \leq \lceil k/r \rceil$ and this is best possible.

The problem naturally extends for sparser host graphs (or hypergraphs). Gyárfás and Lehel conjectured that for k-colored complete bipartite graphs G, $c(G) \leq 2k - 2$ (see [3]), here again $c(G) \leq 2k - 1$ is obvious. For the hypergraph case [5, 6] initiated the study of $c(H)$ when H has bounded independence number.

The main subject of the present paper is the case when the target hypergraph K is a complete r-uniform r-partite hypergraph, i.e., when $V = V(K)$ is partitioned into nonempty classes $V_1 \cup \ldots \cup V_r$ and the edges of K are the sets containing one vertex from each class. Let $\mathrm{cov}(r, k)$ denote the maximum of $c(K)$ when K ranges over spanning k-colorings of complete r-uniform r-partite hypergraphs, and $\mathrm{COV}(r, k)$ denote the maximum of $c(K)$ when K ranges over (not necessarily spanning) k-colorings of complete r-uniform r-partite hypergraphs.

Throughout the paper we **always assume** $r \geq 3$. Our introductory theorem shows that only the spanning colorings are the interesting ones. For any positive integer k, we use the standard notation $[k] = \{1, 2, \ldots, k\}$.

Theorem 1 *If $r \geq 3$, then $\mathrm{COV}(r, k) = k$.*

Proof. Let K be a k-edge-colored r-uniform r-partite complete hypergraph. Take an edge e of K. Let $C_1, \ldots, C_\ell$ be the monochromatic components with $|C_i \cap e| \geq r - 1$. As $r > 2$, clearly no two of them have the same color, so $\ell \leq k$. For every vertex $v \in V$ there is an edge $f \ni v$ with $|f \cap e| = r - 1$, so v is covered by one of these components.
For the sharpness let $V_1 = [k]$ and color each edge e by color $e \cap V_1$. □

We remark that if a coloring of the r-uniform r-partite complete hypergraph is spanning, then *all monochromatic components meet every class.* An edge of color i in a k-colored r-uniform hypergraph K is called **essential** if it is not contained in monochromatic components of any color different from i. When $\mathrm{cov}(r, k)$ is studied we may restrict ourselves to colorings having at least one essential edge in every used color, since otherwise a color can be eliminated by recoloring all edges of that color to some other color and the resulting hypergraph would still have a spanning coloring and the same set of (maximal) monochromatic components. This concept is established in [12] and works well in the proof of our initial result.

Theorem 2 $\mathrm{cov}(r, k) = 1$ *for every $r \geq 3$ and every $1 \leq k \leq r$.*

Proof. Let $e = \{v_1, \ldots, v_r\}$ be an essential edge of color 1 in a complete r-uniform r-partite hypergraph with vertex set $V = \cup_{i=1}^{r} V_i$ where $v_i \in V_i$. Let $R_i = e - \{v_i\}$ and denote by $\mathrm{Col}(R_i) \subseteq [k]$ the set of colors appearing on any edge of the form $R_i \cup \{v_i'\}$ (where $v_i' \in V_i$). Observe that $M = \mathrm{Col}(R_i) \cap \mathrm{Col}(R_j) = \{1\}$ for $i \neq j$. Indeed, $1 \in M$ because e is of color 1 and if $c \in M$ then e is contained in the union of two edges of color c, contradicting to the fact that e is essential. By the pigeonhole principle there exists j such that $\mathrm{Col}(R_j) = \{1\}$. Now V_j is covered by the monochromatic component containing e (of color 1), and, as the coloring is spanning, it necessarily covers the whole V. □

By Theorem 2 from this point we may assume that $k = r + t$ with some integer $t \geq 1$.

Conjecture 3 $\mathrm{cov}(r, r + t) = t + 1$ *for every $r \geq 3$ and every $t \geq 1$.*

It is worth formulating this conjecture in dual form. Assume K is a complete r-uniform r-partite hypergraph with a spanning k-coloring. Consider a new hypergraph H with vertex set $V(K)$ whose edges are the vertex sets of the monochromatic components in the coloring. The dual F (obtained by interchanging vertices and edges and keeping incidences) of this new hypergraph H is a k-uniform k-partite hypergraph whose edges are partitioned into r classes with the property that any r edges from different partite classes have nonempty intersection. As the coloring of K was spanning, monochromatic components have at least r vertices. In this setting Conjecture 3 can be stated in terms of the transversal number $\tau(F)$, the minimum number of vertices intersecting all edges of F.

Conjecture 4 *Assume that the edges of a k-uniform k-partite hypergraph F with minimum degree at least $r \geq 3$ are partitioned into r classes so that any r edges from different classes have nonempty intersection. Then $\tau(F) \leq k - r + 1$.*

In Section 2 we show that Conjecture 3 (if true) is best possible, and it is "almost" true, i.e., $\mathrm{cov}(r, r + t) \leq t + 2$ for every $t \geq 1$ (Theorem 10). We also prove that the conjecture is true for $1 \leq t \leq r - 2$ (Theorem 9). Our most difficult result makes one further step, proving Conjecture 3 for $t = r - 1$ (Theorem 11).

In Section 3 we investigate $c(H)$ for hypergraphs "between" complete and complete partite, in order to build a bridge between the results proved in Section 2 and the results of [12]. We call a hypergraph (r, ℓ)-*partite* if its vertex set is partitioned into r nonempty classes, such that the intersection of any edge

and any class has at most ℓ vertices. We call a hypergraph *complete r-uniform (r,ℓ)-partite* if it contains all r-element sets as edges which meet every partition class in at most ℓ vertices. Let $\mathrm{cov}(r,\ell,k)$ denote the minimum number of monochromatic components needed to cover the vertex set of any complete r-uniform (r,ℓ)-partite hypergraph in any spanning k-coloring. For $2 \le \ell \le r$ we determine exactly the values of $\mathrm{cov}(r,\ell,k)$. We conclude our paper by summarizing the results achieved. Our main result is Theorem 22, stating that

$$\mathrm{cov}(r,\ell,k) = 1 + \left\lfloor \frac{k-r+\ell-1}{\ell} \right\rfloor$$

for every $r \ge 3,\ k \ge 1+r-\ell,\ 1 \le \ell \le r$, *except* for the cases ($\ell = 1$ and $k \ge 2r$), where we could only prove a slightly weaker upper bound.

In the last two sections we examine other generalizations of IRC.

Let K be a k-edge-colored complete graph. Sometimes it is more convenient to work with a color-transitive closure. Here a set of colors $\mathrm{Col}(xy) \subseteq [k]$ are assigned to every edge xy of a complete graph K in such a way, that for three different vertices x,y,z, if $i \in \mathrm{Col}(xy) \cap \mathrm{Col}(yz)$, then $i \in \mathrm{Col}(xz)$ (in other words we put color i into set $\mathrm{Col}(xy)$ if x and y are in the same monochromatic component of color i). Now K_i consists of the edges xy where $i \in \mathrm{Col}(xy)$, and the components of K_i are cliques for all i. On the other hand the set of monochromatic components did not changed. We call K a multi k-edge-colored complete graph if the above transitivity is satisfied, and we use the following notation: $t(K) = \min\{|\mathrm{Col}(xy)| \mid x \ne y \in V(K)\}$. Note that we may have a trivial case: if $t(K) = k$, then every monochromatic component is a spanning clique. We call a multi k-edge-colored complete graph nontrivial if $t(K) < k$.

We conjecture that in a nontrivial multi k-edge-colored complete graph K, the vertex set can be covered by $k - t(K)$ monochromatic components. We prove this conjecture for the case $t(K) > k/4$.

A 3-edge-colored K_4 shows that in IRC sometimes we need at least two components colored by the same color. Motivated by this example, we examine what fraction of the vertices can be covered by $k-1$ monochromatic components of *different* colors in a k-edge-colored complete graph. We prove a sharp bound for this problem, moreover, we also prove that whenever it is sharp, it "comes from" a finite affine plane.

We omit most of the proofs from this extended abstract. For all the missing proofs please see [10] and [13].

2 Results for complete r-uniform r-partite hypergraphs

2.1 Lower bound

Construction 1 *For $t \ge 1,\ r \ge 3,\ k = r+t$, we define a complete r-uniform r-partite hypergraph $K(r,t)$ with a k-coloring of its edges as follows. The vertex set V of $K(r,t)$ is partitioned into r classes, $V_1,\dots,V_r$. The first class V_1 has $\binom{k}{t}$ vertices associated to the t-element subsets of $[k]$. For $2 \le j \le r$ set $V_j = A_j^1 \cup \dots \cup A_j^k$, where the A_j^i-s are disjoint and have $\binom{k-1}{t-1}$ vertices. Fix an arbitrary linear order on every A_j^i.*

First we define special edges of color i for any $i \in [k]$. Consider the set W_i of $\binom{k-1}{t-1}$ vertices of V_1 associated to t-sets of $[k]$ containing i.

- Special edges of color i *are the $\binom{k-1}{t-1}$ edges whose vertex from W_i is the ℓ-th in the standard lexicographic order, and for all $2 \le j \le r$ whose vertex from V_j is the ℓ-th in the fixed linear order of A_j^i for $\ell = 1,\dots,\binom{k-1}{t-1}$. Thus special edges of color i form a matching for all i, $i = 1,\dots,k$.*
- Non-special edges *with vertices $v_1 \in V_1,\dots,v_r \in V_r$ get their color as the smallest $c \in [k]$ such that c is not in the set associated to v_1 and $v_j \notin A_j^c$ for all $2 \le j \le r$.*

Note that every non-special r-tuple $v_1,\dots,v_r$ gets a color because the conditions forbid at most $t+r-1$ colors. Observe also that a special edge of color i is always disjoint from any other edge of color

i. Consequently a special edge of color i forms a monochromatic component of color i having r vertices, we call them *small monochromatic components* and we call any other monochromatic component *large.*

We claim that the given coloring is spanning. Suppose first that $v \in V_1$ representing wlog the set $[t] \subset [k]$. For any $1 \le i \le t$, v is in a special edge of color i. On the other hand, for any $t < i \le r + t$ we can select vertices $v_2 \in A_2^{j_2} \ldots, v_r \in A_r^{j_r}$ so that the upper indices j_t take all values except i from $t + 1, \ldots, t + r$. Then the non-special edge $v, v_2, \ldots, v_r$ is colored by i.

On the other hand, let $v \in A_j^i$ for some $1 < j \le r$, $1 \le i \le k$. Clearly v is in a special edge of color i. For any $c \ne i$ such that $1 \le c \le k$ we can take any vertex $w \in V_1$ associated to a t-set A of $[k]$ such that $c, i \notin A$. Set $B = [k] \setminus (A \cup \{i\} \cup \{c\})$. Then from the $(r-2)$ V_t-s where $t \notin \{1, j\}$ we can pick a set of $r - 2$ vertices with distinct superscripts in B. These vertices together with v, w define an edge that must be colored with c. Thus the coloring of $K(r, t)$ is spanning.

Theorem 5 $\mathrm{cov}(r, r + t) \ge t + 1$ *for every* $r \ge 3$, $t \ge 1$.

Proof. Consider the hypergraph $K(r, t)$. Note that the union of at most t large monochromatic components do not cover the whole V_1. Let their colors be $c_1, \ldots, c_s$ with $s \le t$, and take any t-set that contains $\{c_1, \ldots, c_s\}$; the vertex in V_1 associated to this set is not covered by those large components.

The vertices of V_1 uncovered by the large components must be covered by small monochromatic components, and every such component can contain just one vertex of V_1. Therefore we need $\binom{k-s}{t-s} > t-s$ small monochromatic components to cover them. Thus altogether we need more than $s + (t - s) = t$ monochromatic components to cover the vertices of $K(r, t)$. □

2.2 Upper bounds

We need some additional notation. We assign vectors of length k to every element of the base set $V = V_1 \cup \ldots \cup V_r$. For $v \in V$ the ith coordinate $\mathbf{v}(i)$ of the associated vector $\mathbf{v}$ is the serial number of the monochromatic component of color i containing v. The Hamming distance of two vertices $\delta(v, w) = \delta(\mathbf{v}, \mathbf{w})$ is the number of places the two associated vectors differ.

Statement 6 *For* $i = 1, \ldots, r$ *let* $v_i \in V_i$. *Then there exists* $c \in [k]$ *and an integer* s, *such that* $\mathbf{v}_i(c) = s$ *for all* $i \le r$.

Proof. The edge $e = \{v_1, v_2, \ldots, v_r\}$ is colored by a color, say, by color c. Then the vertices of e belong to the same monochromatic component of color c. □

Lemma 7 *Either* $\mathrm{cov}(r, r + t) = 1$, *or for any two vertices* v, w *from different classes,* $\delta(v, w) \le t + 1$.

Proof. Assume for contradiction that $\delta(v, w) > t + 1$. Wlog $v \in V_1$, $w \in V_2$ and $\mathbf{v} = 1 \ldots 1$ and $\mathbf{w} = 1 \ldots 12 \ldots 2$, where the number of ones is at most $r - 2$. As the coloring is spanning and no monochromatic component covers V, we can choose $v_3, \ldots, v_r$, such that $v_i \in V_i$ and $\mathbf{v}_i(i - 2) > 1$. However, this contradicts Statement 6. Thus the number of twos in $\mathbf{w}$ is at most $t+1$, so $\delta(v, w) \le t+1$. □

Lemma 8 *If* $\mathrm{cov}(r, r + t) > 1$ *and* $W = \{w_1, \ldots, w_\ell\}$ *is a set of vertices from different classes, then for* $J = \{j \in [k] \mid \mathbf{w}_1(j) = \mathbf{w}_2(j) = \ldots = \mathbf{w}_\ell(j)\}$ *we have* $|J| \ge r + 1 - \ell$.

Proof. If $\ell = r$, then this statement coincides with Statement 6. Otherwise suppose indirectly that $|J| \le r-\ell$, and $J = \{j_1, \ldots, j_{|J|}\}$. We may choose $|J| \le r-\ell$ vertices $u_1, \ldots, u_{|J|}$ from different classes that do not contain any $w_j \in W$ so that $\mathbf{u}_i(j_i) \ne \mathbf{w}_1(j_i)$. If $|J| < r - \ell$, we can extend $w_1, \ldots, w_\ell, u_1, \ldots, u_{|J|}$ with vertices from new vertex classes to obtain r vertices and defining an edge that contradicts Statement 6. □

Using these lemmas, the following theorems can be proved.

Theorem 9 $\mathrm{cov}(r, r + t) \le t + 1$ *for every* $1 \le t \le r - 2$ *and* $r \ge 3$.

Theorem 10 $\mathrm{cov}(r, r + t) \le t + 2$ *for every* $2 \le r - 1 \le t$.

2.3 The case $t = r - 1$

Our most difficult result makes one further step, proving Conjecture 3 for $t = r - 1$.

Theorem 11 $\mathrm{cov}(r, 2r-1) = r$ *if* $r \geq 3$.

Suppose the statement does not hold, let $k = 2r-1$ and fix a k-colored r-uniform r-partite hypergraph K where $c(K) \geq r+1$ (and the coloring is spanning). We proved the following subsequent statements.

Claim 12 *For any* $i \neq j$ *and* $a \in V_i, b \in V_j$ *we have*

$$r - 1 \leq \delta(a, b) \leq r.$$

Claim 13 *If* a, b *are two vertices from different partite classes such that* $\delta(a, b) = r - 1$*, then some of these classes contain two vertices with Hamming distance* $2r - 1$.

Claim 14 *For any two vertices* v, w *from the same partite class,*

$$\delta(\mathbf{v}, \mathbf{w}) < 2r - 1.$$

Combining the claims we conclude with the following corollary.

Corollary 15 *For any two vertices* v, w *from different classes,* $\delta(v, w) = r$*, and for any two vertices* u, v *from the same class,* $\delta(u, v) \leq 2r - 2$.

Using these statements, Theorem 11 can be proved (see [10] for the details). Summarizing the results achieved so far:

Corollary 16 *If* $r \geq 3$*, then* $\mathrm{cov}(r, k) = 1$ *for every* $1 \leq k \leq r$,
$\mathrm{cov}(r, k) = k - r + 1$ *for every* $r \leq k \leq 2r - 1$,
and for any $k \geq 2r$ *we have* $k - r + 1 \leq \mathrm{cov}(r, k) \leq k - r + 2$.

3 Generalized complete uniform hypergraphs

Definition 17 *A hypergraph is called* (r, ℓ)-**partite** *if the ground set* V *is partitioned into nonempty classes* $V_1 \cup \ldots \cup V_r$*, and no edge intersects any* V_i *in more than* ℓ *vertices. A hypergraph is complete* r*-uniform* (r, ℓ)*-partite if its edge set consists of all* r*-tuples intersecting each class in at most* ℓ *vertices. An edge of an* (r, ℓ)*-partite hypergraph is called* friendly *if it intersects at most one class in exactly* ℓ *vertices; otherwise we call it* unfriendly. *An* r*-uniform* (r, ℓ)*-partite hypergraph is called* **semicomplete** *if its edge set consists of all* r*-tuples intersecting at most one class in exactly* ℓ *vertices (that is, it consists of the friendly edges of the complete* r*-uniform* (r, ℓ)*-partite hypergraph). An* r*-uniform* (r, ℓ)*-partite hypergraph is called* **rich** *if it contains all edges of the semicomplete hypergraph.*

Among r-uniform hypergraphs the complete $(r, 1)$-partite hypergraphs are the complete r-partite ones and complete (r, r)-partite hypergraphs are the complete ones. The complete $(r, r-1)$-partite hypergraphs are also interesting, containing all r-tuples of V except those that are contained in some V_i. The purpose of this section is to build a bridge between the two known extreme cases ($\ell = r$ was solved in [12], $\ell = 1$ was handled in the previous section).

For $1 \leq \ell \leq r$, let $\mathrm{cov}(r, \ell, k)$ denote the minimum number of monochromatic components needed to cover the vertex set of any complete r-uniform (r, ℓ)-partite hypergraph in any spanning k-coloring.

Conjecture 18

$$\mathrm{cov}(r, \ell, k) = 1 + \left\lfloor \frac{k - r + \ell - 1}{\ell} \right\rfloor$$

for every $r \geq 3,\ k \geq 1 + r - \ell,\ 1 \leq \ell \leq r$.

We start with giving the lower bound.

Construction 2 *This construction is a straightforward generalization of Construction 1. We have r, k, ℓ fixed with $k \geq r+1 \geq 4$ and $1 \leq \ell \leq r$, and let $q = \lfloor \frac{k-r+\ell-1}{\ell} \rfloor$ and $k' = q \cdot \ell + r - \ell + 1 \leq k$. First we fix the sizes and labels of the classes. $|V_1| = \binom{k'}{q}$ and elements V_1 are labeled with the q-element subsets of $[k']$. For $2 \leq j \leq r$ define V_j as $V_j = A_j^1 \cup \cdots \cup A_j^{k'}$ where the sets in the union are disjoint, $|A_j^i| = \binom{k'-1}{q-1}$, and all elements of A_j^i are labeled with set $\{i\}$ and have an arbitrary fixed linear order. Now take an arbitrary rich r-uniform (r,ℓ)-partite hypergraph H_{rich} on $V = V_1 \cup \ldots \cup V_r$, we are going to define a spanning k'-coloring of its edges.*

First we define special edges of color i for any $i \in [k']$. Consider the set W_i of $\binom{k'-1}{q-1}$ vertices of V_1 associated to q-sets of $[k']$ containing i.

Special edges of color i *are the $\binom{k'-1}{q-1}$ edges whose vertex from W_i is the ℓ-th in the standard lexicographic order, and for all $2 \leq j \leq r$ whose vertex from V_j is the ℓ-th in the fixed linear order of A_j^i for $\ell = 1, \ldots, \binom{k'-1}{q-1}$. Thus special edges of color i form a matching for all i.*

Non-special edges *with vertices $v_1, \ldots, v_r$ get their color as the smallest $c \in [k']$ such that c is not in the union of sets associated to $v_1, \ldots, v_r$.*

Note that every non-special r-tuple $v_1, \ldots, v_r$ gets a color because the conditions forbid at most $\ell \cdot q + (r-\ell) < k'$ colors. Observe also that a special edge of color i is always disjoint from any other edge of color i. Consequently a special edge of color i forms a monochromatic component of color i having r vertices, we call them small monochromatic components.

We claim that the coloring is spanning. Suppose first that $v_1 \in V_1$ representing the set $Q_1 \subset [k']$. For any $i \in Q_1$, v_1 is in a special edge of color i. On the other hand, for any $i \notin Q_1$ we can select vertices $v_2, \ldots, v_\ell \in V_1$ with associated q-sets $Q_2, \ldots, Q_\ell \subseteq [k'] - \{i\}$, such that for every $j \neq j'$ sets Q_j and $Q_{j'}$ are disjoint. Then we may select $v_{\ell+1}, \ldots, v_r$ from $V_2, \ldots, V_{r-\ell+1}$, such that the associated one-element subsets are distinct, and are subsets of $[k'] - \{i\} - \cup Q_j$. Now the union of the associated sets of our selected r-tuple is $[k'] - \{i\}$, thus it was colored by i.

On the other hand, let wlog $v_r \in A_r^i$ for some $1 \leq i \leq k'$. Clearly v_r is in a special edge of color i. For any $1 \leq c \leq k'$ if $c \neq i$, then we can take vertices $v_1, \ldots, v_\ell \in V_1$ with associated q-sets $Q_1, \ldots, Q_\ell \subseteq [k'] - \{i\} - \{c\}$, such that for every $j \neq j'$ sets Q_j and $Q_{j'}$ are disjoint. Then we may select $v_{i+\ell-1} \in V_i$ for $i = 2, \ldots, r-\ell$, such that the associated one-element subsets are distinct, and are subsets of $[k'] - \{i\} - \{c\} - \cup Q_j$. Now the union of the associated sets of our selected r-tuple is $[k'] - \{c\}$, thus it was colored by c.

Theorem 19 $\mathrm{cov}(r, \ell, k) \geq 1 + \lfloor \frac{k-r+\ell-1}{\ell} \rfloor$ *for every $r \geq 3$, $k \geq 1 + r - \ell$, $1 \leq \ell \leq r$.*

Proof. The statement is obvious if $k \leq r$. Consider Construction 2. Note that the union of at most $q = \lfloor \frac{k-r+\ell-1}{\ell} \rfloor$ large monochromatic components do not cover V_1. Let their colors be $c_1, \ldots, c_s$ with $s \leq q$, and take any q-set that contains $\{c_1, \ldots, c_s\}$; the vertex in V_1 associated to this set is not covered.

The uncovered vertices of V_1 must be covered by small monochromatic components, and every such component can contain just one vertex of V_1. Therefore we need $\binom{k'-s}{q-s} > q - s$ small monochromatic components to cover them, thus altogether we need more than $s + (q-s) = q$ monochromatic components to cover all vertices. □

Remark 20 *The basic idea of the above construction is from [12] where the constructed coloring for complete r-uniform hypergraphs is not spanning (this was not an issue of that paper). Here, when $\ell = r$, we gave another construction for complete r-uniform hypergraphs where the coloring is spanning.*

Theorem 21 $\mathrm{cov}(r, \ell, k) \leq 1 + \lfloor \frac{k-r+\ell-1}{\ell} \rfloor$ *for every $r \geq 3$, $k \geq 1 + r - \ell$, $2 \leq \ell \leq r$.*

Proof. The proof goes similarly as in the proof of Theorem 2. Fix the nonempty classes $V_1, \dots, V_r$ and take any rich r-uniform (r,ℓ)-partite hypergraph H_{rich} with a spanning k-coloring of its edges. We are going to show by induction on k that $c(H_{\text{rich}}) \le 1 + \lfloor \frac{k-r+\ell-1}{\ell} \rfloor$. The cases $k \le r$ are obvious.

Let $e = \{u_1, \dots, u_r\}$ be an essential edge of H_{rich} colored by 1, if no such edge exists, then recolor edges having color 1 and use induction. Until there exists an essential friendly edge colored by 1, we choose that edge for e. If all essential edges colored by 1 are unfriendly, then simply delete them from H_{rich} getting a $(k-1)$-colored rich hypergraph, where the coloring is still spanning, so we are done by induction.

So e is a friendly essential edge, wlog $\ell \ge |e \cap V_1| \ge |e \cap V_j|$ for all j. As e is friendly, we also have $|e \cap V_j| < \ell$ for $j > 1$. Take $R_{u_1}, \dots, R_{u_r}$, where $R_{u_j} = e - \{u_j\}$, for any $i \ne j$ we have $\text{Col}(R_{u_i}) \cap \text{Col}(R_{u_j}) = \{1\}$, so there is a j with $|\text{Col}(R_{u_j})| \le 1 + \lfloor \frac{k-1}{r} \rfloor$.

First consider the case $|R_{u_j} \cap V_1| < \ell$ (note that this is always true for $\ell = r$). We also emphasize here that for this case we do not need the coloring to be spanning. For any vertex $v \in V$ the set $R_{u_j} \cup \{v\}$ is a friendly edge of H_{rich}, consequently the monochromatic components of colors in $\text{Col}(R_{u_j})$ containing R_{u_j} cover the whole V. We need to prove $\lfloor \frac{k-1}{r} \rfloor \le \lfloor \frac{k-r+\ell-1}{\ell} \rfloor$. For $k-1 < r$ both are zero, otherwise $(r-\ell)(k-1) \ge (r-\ell)r$, so $\frac{k-1}{r} \le \frac{k-r+\ell-1}{\ell}$.

So we are left with the case $|R_{u_j} \cap V_1| = \ell$. There are two possibilities. Either one of $\text{Col}(R_{u_i}) = \{1\}$ for an $i > 1$, in this case the monochromatic component containing u_i and colored by 1 covers V because it covers $V - V_1$, as for all $v \in V - V_1$ the set $e - \{u_i\} \cup \{v\}$ is an edge of H_{rich}, and (using that the coloring is spanning), every $w \in V_1$ is incident to an edge colored by 1 and this edge meets $V - V_1$.

Otherwise $|\text{Col}(R_{u_i})| \ge 2$ for all $i > 1$, so by the pigeonhole principle there is a $2 \le i \le \ell$ with $|\text{Col}(R_{u_i})| \le 1 + \lfloor \frac{k-1-(r-\ell)}{\ell} \rfloor$, and the monochromatic components of colors in $\text{Col}(R_{u_i})$ containing $e - \{u_i\}$ cover the whole V because $e - \{u_i\} \cup \{v\}$ is an edge of H_{rich} for every $v \in V - e$. □

Summarizing the results of this section and Corollary 16, we proved Conjecture 18 for almost all cases. We also proved that Conjecture 18 is equivalent to Conjecture 3.

Theorem 22 (Main theorem)

$$\text{cov}(r,\ell,k) = 1 + \left\lfloor \frac{k-r+\ell-1}{\ell} \right\rfloor$$

for every $r \ge 3$, $k \ge 1 + r - \ell$, $1 \le \ell \le r$, except *when* $\ell = 1$ *and* $k \ge 2r$, *where only* $1 + \lfloor \frac{k-r+\ell-1}{\ell} \rfloor \le \text{cov}(r,\ell,k) \le 2 + \lfloor \frac{k-r+\ell-1}{\ell} \rfloor$ *was proved.*

4 The t-intersecting conjecture

Conjecture 23 *Let H be an k-uniform k-partite t-intersecting hypergraph $(1 \le t < k)$. Then $\tau(H) \le k - t$.*

In the dual language it translates as follows. (Remember that $t(K) = \min\{|\text{Col}(xy)| \mid x \ne y \in V(K)\}$.)

Conjecture 24 *Let K be a nontrivial multi k-edge-colored complete graph. Then $V(K)$ can be covered by at most $k - t(K)$ monochromatic components.*

This conjecture is seemingly a generalization of IRC. However, if the statement is proved for $t(K) = \ell - 1 > 0$, then it is also true for $t(K) = \ell$. Suppose we are given a nontrivial multi k-edge-colored complete graph with $t(K) = \ell < k$. Take an edge xy with $\text{Col}(xy) = t(K)$, wlog we may suppose that $k \in \text{Col}(xy)$. Delete color k from the color set of each edge. The resulting complete graph K' is nontrivial multi $(k-1)$-edge-colored and $t(K') = t(K) - 1 = \ell - 1$.

If x is a vertex and $I \subseteq [k]$ is a set of colors, then we denote by $\mathcal{C}(x, I)$ the set of monochromatic components containing x and having a color in I.

Theorem 25 *Let K be a nontrivial multi k-edge-colored complete graph. If $t := t(K) > \frac{k}{4}$, then $V(K)$ can be covered by at most $k - t$ monochromatic components.*

Here we only prove the statement of the theorem for $k \leq 4t - 2$, for the case of $k = 4t - 1$ please see [13].

Proof. Choose an edge xy with $|\mathrm{Col}(xy)| = t$, wlog we can suppose that $\mathrm{Col}(xy) = I = [t]$. Moreover, as the coloring is nontrivial, we have $t < k$. First consider the case $k \leq 2t$. Let $J = [k - t]$, now $J \subseteq I$. We claim that $\mathcal{C}(x, J) = \mathcal{C}(y, J)$ covers $V(K)$. If a vertex z is not covered, then $\mathrm{Col}(xz) = \mathrm{Col}(yz) = \{k-t+1, \ldots, k\}$. However, since each monochromatic component is a clique, we get $\{k-t+1, \ldots, k\} \subseteq I$, so $t = |I| = k$ contradicting to the assumption $t < k$.

Thus we are remained to prove the case $k > 2t$. Let $j = \lfloor \frac{k}{2} \rfloor - t$ and $J = \{t + 1, \ldots, t + j\}$ if $j > 0$ and $J = \emptyset$ otherwise. Take $\mathcal{C}(x, I) \cup \mathcal{C}(x, J) \cup \mathcal{C}(y, J)$. We claim that these $t + 2j \leq k - t$ monochromatic components cover the vertices of K. If a vertex z is not covered, then $\mathrm{Col}(xz) \subseteq \{t + j + 1, \ldots, k\}$ and $\mathrm{Col}(yz) \subseteq \{t + j + 1, \ldots, k\}$ and, as each monochromatic component is a clique, $\mathrm{Col}(xz) \cap \mathrm{Col}(yz) \subseteq I$, so $\mathrm{Col}(xz) \cap \mathrm{Col}(yz) = \emptyset$. However, $|\mathrm{Col}(xz)| \geq t$ and $|\mathrm{Col}(yz)| \geq t$, so $2t \leq k - t - j$, i.e., $2t \leq \lceil \frac{k}{2} \rceil$ or equivalently $k \geq 4t + 1$, a contradiction. □

5 Covering large fraction by few monochromatic components

In this section, we give a sharp bound for the ratio of vertices that can be covered by $k-1$ monochromatic components of pairwise different colors in a multi k-edge colored complete graph.

Theorem 26 *Let K be a multi k-edge-colored complete graph on n vertices. Then at least $\left(1 - \frac{k-2}{(k-1)^2}\right) \cdot n$ vertices of K can be covered by $k - 1$ monochromatic components of pairwise different colors, and this bound is sharp for infinitely many values of k.*

Applying backwards the construction of Gyárfás, we get the following statement for hypergraphs.

Theorem 27 *If H is an k-partite k-uniform intersecting hypergraph, then at least $\left(1 - \frac{k-2}{(k-1)^2}\right) \cdot |E(H)|$ edges of H can be covered by $k - 1$ points from pairwise different classes, and this bound is sharp for infinitely many values of k.*

Characterization of sharp examples

We are able to characterize the sharp examples for Theorem 26. Let us start with a definition.

Definition 28 *We call a multi edge-colored graph K the blowup of an affine plane, if there is an affine plane $\mathcal{A} = (\mathcal{P}, \mathcal{L})$, whose lines are colored such that two lines have the same color if and only if they are disjoint (i.e., parallel), and a positive integer b, such that to every point $p \in \mathcal{P}$ of the affine plane, b vertices correspond in $V(K)$, and two vertices are connected by an edge having color i if and only if the corresponding points in $\mathcal{A}$ are incident to a common line of color i (this includes also the case if the two points correspond to the same point of $\mathcal{A}$).*

Theorem 29 *For a multi k-edge-colored complete graph K on n vertices, the maximum number of vertices coverable by $k - 1$ monochromatic components of pairwise different colors equals $\left(1 - \frac{k-2}{(k-1)^2}\right) \cdot n$ if and only if K is a blowup of an affine plane.*

References

[1] R. Aharoni Ryser's conjecture for tripartite 3-graphs, *Combinatorica* **21** (2001), pp. 1–4.

[2] R. Aharoni, P. Haxell Hall's theorem for hypergraphs, *J. Graph Theory* **35** (2000), pp. 83–88.

[3] G. Chen, S. Fujita, A. Gyárfás, J. Lehel and Á. Tóth, Around a biclique cover conjecture, *arxiv:1212.6861*

[4] P. Erdős, A. Gyárfás and L. Pyber, Vertex coverings by monochromatic cycles and trees, *Journal of Combinatorial Theory B* **51.** (1991) pp. 90–95.

[5] S. Fujita, M. Furuya, A. Gyárfás and Á. Tóth, Partition of graphs and hypergraphs into monochromatic connected parts, *Electronic Journal of Combinatorics* **19** (2012) P27.

[6] S. Fujita, M. Furuya, A. Gyárfás and Á. Tóth, A note on covering edge colored hypergraphs by monochromatic components, *Electronic Journal of Combinatorics* **21** (2014) P33.

[7] Z. Füredi Maximum degree and fractional matchings in uniform hypergraphs, *Combinatorica* **1** (1981), pp. 155–162.

[8] A. Gyárfás, Partition covers and blocking sets in hypergraphs, *MTA SZTAKI tanulmányok* **71** (1977) (in Hungarian)

[9] A. Gyárfás, Vertex covers by monochromatic pieces – A survey of results and problems, *Discrete Mathematics*, **339** (2016) pp. 1970.1977.

[10] A. Gyárfás and Z. Király Covering complete partite hypergraphs by monochromatic components, *Egres Technical Report TR-2016-03*, www.cs.elte.hu/egres/

[11] J. R. Henderson Permutation Decompositions of $(0,1)$-matrices and decomposition transversals, *PhD. Thesis, Caltech* (1971)
thesis.library.caltech.edu/5726/1/Henderson_jr_1971.pdf

[12] Z. Király, Monochromatic components in edge-colored complete hypergraphs, *European Journal of Combinatorics* **35** (2013) pp. 374–376.

[13] Z. Király and L. Tóthmérész On some special cases of Ryser's conjecture, *Egres Technical Report TR-2016-14*, www.cs.elte.hu/egres/

Optimal pebbling and rubbling of graphs with given diameter

Ervin Győri[1]

Alfréd Rényi Institute of Mathematics
1053 Budapest, Reáltanoda utca 13-15.,
Hungary
gyori.ervin@renyi.mta.hu

Gyula Y. Katona[2]

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
1117 Budapest, Műegyetem rkpt. 2, Hungary
and
MTA-ELTE Numerical Analysis and Large
Networks Research Group
kiskat@cs.bme.hu

László F. Papp[3]

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
1117 Budapest, Műegyetem rkpt. 2, Hungary
lazsa@cs.bme.hu

Abstract: A pebbling move on a graph removes two pebbles from a vertex and adds one pebble to an adjacent vertex. A vertex is reachable from a pebble distribution if it is possible to move a pebble to that vertex using pebbling moves. The optimal pebbling number π_{opt} is the smallest number m needed to guarantee a pebble distribution of m pebbles from which any vertex is reachable. A rubbling move is similar to a pebbling move, but it can remove the two pebbles from two different vertex. The optimal rubbling number ρ_{opt} is defined analogously to the optimal pebbling number. In this paper we give lower bounds on both the optimal pebbling and rubbling numbers by the distance k domination number. With this bound we prove that for each k there is a graph G with diameter k such that $\rho_{\text{opt}}(G) = \pi_{\text{opt}}(G) = 2^k$.

Keywords: graph pebbling, rubbling, diameter, distance domination

1 Introduction

Graph pebbling is a game on graphs initialized by a question of Saks and Lagarias, which was answered by Chung in 1989 [3]. Its roots are originated in number theory.

Each graph in this paper is simple and connected. We denote the vertex set and the edge set of graph G with $V(G)$ and $E(G)$, respectively. The distance between vertices u and v, denoted by $d(u, v)$, is the minimum number of edges contained in the shortest path connecting u and v. We use $\text{diam}(G)$ for the diameter of G.

[1]Supported by the National Research, Development and Innovation Office NKFIH, No. 116769.
[2]Supported by the National Research, Development and Innovation Office NKFIH, No. 116769 and No. 108947.
[3]Supported by the National Research, Development and Innovation Office NKFIH, No. 108947.

We write $G\Box H$ for the Cartesian product of graphs G and H. The vertex set of $G\Box H$ is $V(G)\times V(H)$ and vertices (g,h) and (g',h') are adjacent if and only if either $g=g'$ and $\{h,h'\}\in E(H)$, or $h=h'$ and $\{g,g'\}\in E(G)$. We use $G^{\Box d}$ as an abbreviation of $G\Box G\Box\ldots\Box G$ where G appears exactly d times.

A *pebble distribution* P on graph G is a function mapping the vertex set to nonnegative integers. We can imagine that each vertex v has $P(v)$ pebbles. A *pebbling move* removes two pebbles from a vertex and places one to an adjacent one. A pebbling move is *allowed* if and only if the vertex loosing pebbles has at least two pebbles.

A sequence of pebbling moves is called *executable* if for any i the ith move is allowed under the the distribution obtained by the application of the first $i-1$ move. The pebble distribution which we get from P after the execution of the sequence of pebbling moves σ is denoted by P_σ.

A vertex v is *reachable* under a distribution P, if there is an executable sequence of pebbling moves σ, such that $P_\sigma(v)\geq 1$. We say that a distribution P is *solvable* if each vertex is reachable under P. The size of a pebble distribution P is $\sum_{v\in V(G)}P(v)$ which we denote by $|P|$. A pebble distribution P on a graph G will be called *optimal* if it is solvable and its size is the smallest possible. The size of an optimal pebble distribution is called *the optimal pebbling* number and denoted by $\pi_{opt}(G)$.

In [1] the authors invented a version of pebbling called rubbling. The only difference between the definitions of pebbling and rubbling is that there is an additional available move. A *strict rubbling move* removes two pebbles in total but it takes them from two different vertices then it places one pebble at one of their common neighbours. Thus a strict rubbling move is allowed if it removes pebbles from vertices who share a neighbour and both of them has a pebble. A rubbling move is either a pebbling move or a strict rubbling move. If we replace pebbling moves with rubbling moves everywhere in the definition of the optimal pebbling number, then we obtain the *optimal rubbling number*, which is denoted by ρ_{opt}.

There are not many results on rubbling, only two articles [10, 11] appeared about rubbling so far. On the other hand, the optimal pebbling number of several graph families are known. For example exact values were given for paths and cycles [6, 14], ladders [2], caterpillars [4], m-ary trees [5] and staircase graphs [8]. However, determining the optimal pebbling number for a given graph is NP-hard [12]. There are also some known bounds on the optimal pebbling number. One of the earliest is that $\pi_{\mathrm{opt}}(G)\leq 2^{\mathrm{diam}(G)}$.

Placing $2^{\mathrm{diam}(G)}$ pebbles to a single vertex always creates a solvable distribution, but usually much less pebbles are enough to construct a solvable distribution. It is a natural question, if there are graphs with arbitrary large diameter where this amount of pebbles is required for an optimal pebbling?

The answer is positive and it was given in [13]. However, the proof in [13] is incorrect. The authors gave a set of graphs and claimed that they have this property, but we will show during the proof of Claim 1 that it is not true.

Herscovici *et al.* in [9] proved that $\pi_{\mathrm{opt}}(K_m^{\Box d})=2^d$ if $m>2^{d-1}$. In fact, a more general statement is proved in [9], but this is enough for our purposes. The diameter of these graphs is d, therefore they prove the sharpness of the diameter bound.

We can ask, what happens when we consider rubbling instead of pebbling? Unfortunately the proof of Herscovici *et al.* rely on several phenomena true for pebbling but false for rubbling. We answer this question and prove that $\rho_{\mathrm{opt}}(K_m^{\Box d})=2^d$ if $m\geq 2^d$. Since $\rho_{\mathrm{opt}}(G)\leq\pi_{\mathrm{opt}}(G)$, it is also a new short proof for the pebbling case. Our method uses the concept of distance domination.

A distance k domination set S of a graph is a subset of the vertex set such that for each vertex v there is an element s of S whose distance from v is at most k. The distance k domination number of a graph, denoted by γ_k, is the size of the smallest distance k domination set.

First we prove that $\rho_{\mathrm{opt}}(G)\geq\min\left(\gamma_{k-1}(G),2^k\right)$ for each k, then we give an improved lower bound using both γ_{k-1} and γ_{k-2}.

Finally we use these bounds to show that $\pi_{\mathrm{opt}}(K_3\Box K_3\Box K_5)=6$.

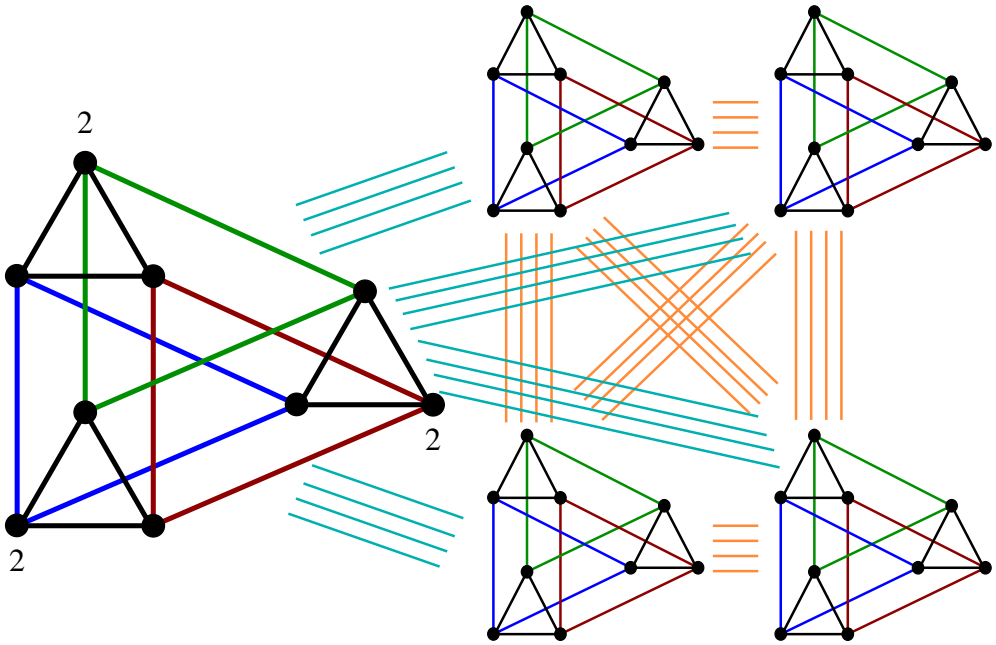


Figure 1: An optimal distribution of $K_3 \square K_3 \square K_5$ using 6 pebbles.

2 Main result

2.1 Counterexample to the proof of Muntz *et al.*

Muntz *et al.* give an iterative construction of graphs. They claim in [13], that if G is a graph with diameter d and its optimal pebbling number is 2^d, then $G \square K_{2^d+1}$ is a graph with diameter $d+1$ and optimal pebbling number 2^{d+1}. It is easy to see that $\operatorname{diam}(G \square K_{2^d+1}) = d+1$, however its optimal pebbling number is not necessarily $2\pi_{\text{opt}}(G)$.

Muntz *et al.* choose K_3 as a starting graph. The third graph in the sequence is $K_3 \square K_3 \square K_5$. The optimal pebbling number of this graph is not 8, as the authors claimed.

Claim 1 *The optimal pebbling number of $K_3 \square K_3 \square K_5$ is at most* 6.

Proof: A solvable distribution with 6 pebbles is given in Figure 1. We can move two pebbles to each vertex of the leftmost $K_3 \square K_3$. Since each other vertex is connected to these vertices, all vertices are reachable. □

Furthermore, all later graphs in the sequence are counterexamples. Because if we take a solvable distribution of G and use the double of its pebbles on a copy of G in $G \square K_n$, then we get a solvable distribution of $G \square K_n$. Thus if $\pi_{opt}(G) < 2^d$, then $\pi_{opt}(G \square K_n) < 2^{d+1}$.

Besides, it can be proven that changing the starting graph does not help, the construction fails.

2.2 A lower bound given by the distance domination number

We establish our first lower bound on the optimal pebbling and rubbling numbers using the distance k domination number.

Theorem 2 *Let G be a connected graph and k be an integer greater than one, then:*

$$\rho_{\text{opt}}(G) \geq \min\left(\gamma_{k-1}(G), 2^k\right).$$

The *support* of a pebble distribution P, denoted by $\mathrm{supp}(P)$ is the set of vertices containing at least one pebble.

The *weight function* of P, which is defined on the vertex set of G, is $W_P(u) = \sum_{v \in V(G)} P(v) 2^{-d(u,v)}$.

Clearly, if a vertex is reachable under P, then its weight is at least one. This is true for both pebbling and rubbling.

PROOF: Consider a pebble distribution P whose size is less than both $\gamma_{k-1}(G)$ and 2^k. Hence $\mathrm{supp}(P)$ is not a distance $k-1$ dominating set. There is a vertex v whose distance from $\mathrm{supp}(P)$ is k. Therefore the weight of this vertex is $\frac{1}{2^k}|P| < 1$, hence v is not reachable under P. So a solvable pebble distribution has at least $\min(\gamma_{k-1}(G), 2^k)$ pebbles. □

We are free to choose k. The best bound is obtained when $\gamma_{k-1} \approx 2^k$.

Notice that the proof exploited that each vertex contains integer pebbles and that the degradation of pebbles is exponential in sense of the distance. On the other hand, we have not used that a pebbling or a rubbling move removes integer number of pebbles. Therefore this method also works when a pebble can be broken to arbitrary small pieces. Hence it also gives a bound on the optimal integer fractional covering ratio which is defined in [7].

2.3 The optimal rubbling number of $K_m^{\Box d}$ is 2^d if $m \geq 2^d$

Let $\Sigma_{m,k}$ be the following graph: We choose an alphabet Σ of size m. The vertices of $\Sigma_{m,k}$ are the words over Σ of length k. Two vertices are adjacent if and only if the corresponding words differ only at one position, roughly speaking their Hamming distance is one. It is well known that $\Sigma_{m,k} \simeq K_m^{\Box k}$. We use this coding theory approach because it is more natural for us to interpret the following proofs in this language.

It is easy to see that $\mathrm{diam}(\Sigma_{m,k}) = k$: We have to change all k characters of word $a, a, \ldots, a$ to obtain $b, b, \ldots, b$, each of the changes requires passing through an edge. We can obtain any word from any other by changing each character at most once, hence $\mathrm{diam}(\Sigma_{m,k}) = k$.

Claim 3 $\gamma_{k-1}(\Sigma_{m,k}) = m$.

PROOF: The set containing all constant words over alphabet Σ with length k is a distance $k-1$ dominating set, because it is enough to change at most $k-1$ characters of a k long word to obtain a constant one. The number of these words is m.

Let A be a set of words over alphabet Σ with length k such that the size of A is $m-1$. Consider the ith characters of all words contained in A. The pigeonhole principle implies that there is a character $c_i \in \Sigma$ which does not appear among them. Such a character exists for each position. Consider the word $c = c_1 c_2 \ldots c_k$. We have to change all of its characters to obtain a word contained in A, thus its distance from A is k so A is not a distance $k-1$ dominating set. □

Theorem 4 *Both the optimal pebbling and optimal rubbling number of $K_m^{\Box d}$ is 2^d if $m \geq 2^d$.*

PROOF: We have already seen that 2^d pebbles at a single vertex is enough to construct a solvable pebble distribution even if we consider only pebbling moves.

For the lower bound we set k as d and apply Theorem 2. The obtained lower bound is also 2^d. □

2.4 Lower bounds using both γ_{k-1} and γ_{k-2}

To improve Theorem 2, we have to use several properties of pebbling and rubbling. Therefore the obtained bounds are no longer the same for π_{opt} and ρ_{opt}.

Let S be a subset of $V(G)$. The open neighbourhood of vertex v is the set of vertices which are adjacent to v. The closed neighbourhood contains the adjacent vertices plus the vertex itself. The closed

neighbourhood of set S, denoted by $N[S]$, is defined as the union of the closed neighbourhoods of vertices contained in S. We write $N(S)$ for the open neighbourhood of S which is defined as $N[S] \setminus S$.

Let σ be a sequence of pebbling moves and let M be a pebbling move which contained in σ. We write $\sigma \setminus M$ for the sequence of pebbling moves which we get after we delete the last appearance of M. If we add an additional pebbling move T to the beginning of σ, then we denote the obtained sequence by $T\sigma$.

Let P be a pebble distribution on G and S be an arbitrary subset of $V(G)$. Then the the *restriction* of P to S is a pebble distribution which is defined as follows:

$$P|_S = \begin{cases} P(v) & \text{if } v \in S \\ 0 & \text{otherwise} \end{cases}$$

Theorem 5 *For all $k \geq 3$ and any graph G whose edge set is non empty we have:*

$$\pi_{\text{opt}}(G) \geq \min\left(2^k, \gamma_{k-1}(G) + 2^{k-2}, \gamma_{k-2}(G) + 1\right)$$

Proof: Consider a solvable pebble distribution P. We have already seen that $|P| \geq \min(\gamma_{k-1}(G), 2^k)$.

Assume that $|P| < \min(\gamma_{k-2}(G) + 1, 2^k)$. Either $\text{supp}(P)$ is not a distance $k-2$ domination set or each vertex has at most one pebble. In the later case there are no available pebbling moves but there are vertices which do not have pebbles, so they are not reachable which is a contradiction.

In the other case, there is a vertex v whose distance from $\text{supp}(P)$ is at least $k-1$. On the other hand, $\text{supp}(P)$ has to be a distance $k-1$ domination set, since otherwise 2^k pebbles would be required to reach some of the vertices.

Let σ be an executable sequence of pebbling moves moving a pebble to v. We say that a subdivision of σ to two subseqences τ and μ is proper if τ and μ are executable under P and P_τ, respectively and μ does not contain a move which removes a pebble from $\text{supp}(P)$. We chose a proper subdivision where the size of μ is maximal.

We execute τ and investigate the obtained distribution P_τ. We show that $\text{supp}(P_\tau) \subseteq N[\text{supp}(P)]$: Assume that a vertex outside of $N[\text{supp}(P)]$ has a pebble under P_τ. Then the last pebbling move M which placed it there does not remove pebbles from $\text{supp}(P)$. $\tau \setminus M$ is executable and if we put M to the beginning of μ then $M\mu$ is also executable under $P_{\tau \setminus M}$. Furthermore $M\mu$ does not remove a pebble from $\text{supp}(P)$, thus $\tau \setminus M$, $M\mu$ is a proper subdivision of σ which contradicts with the maximality of μ. Therefore $\text{supp}(P_\tau) \subseteq N[\text{supp}(P)]$.

At each vertex of $\text{supp}(P)$ the execution of τ either leaves a pebble or it removes at least two pebbles by a pebbling move which consumes one pebble. Thus at most $|P| - |\text{supp}(P)|$ pebbles arrive at $N(\text{supp}(P))$ after the execution of τ.

μ uses only these pebbles and moves a pebble to v. Therefore v is reachable under $P_\tau|_{N(\text{supp}(P))}$. The distance of v from $\text{supp}(P_\tau)$ is at least $k-2$, therefore $2^{2-k}(|P| - |\text{supp}(P)|) \geq W_{P\tau|N(\text{supp}(P))}(v) \geq 1$. Since $|\text{supp}(P)| \geq \gamma_{k-1}(G)$, we get that $|P| \geq 2^{k-2} + \gamma_{k-1}(G)$.

So either $|P| \geq 2^{k-2} + \gamma_{k-1}(G)$ or our assumption was false and $|P| \geq \min(\gamma_{k-2}(G)+1, 2^k)$. Altogether these imply the desired result. □

If we talk about rubbling, then there are two main differences. First, a distribution which places at most one pebble everywhere and leaving a vertex without a pebble can be solvable. Second, a rubbling move can remove pebbles from two vertices and consume just one pebble, hence we can state just that $|P| - \frac{|\text{supp}(P)|}{2}$ pebbles arrive at $N(\text{supp}(P))$ after the execution of τ. If we change the above proof accordingly, then we get the following improved version of Theorem 2 for rubbling:

Theorem 6 *For all $k \geq 2$ and all graphs G we have:*

$$\rho_{\text{opt}}(G) \geq \min\left(2^k, \max\left(\frac{\gamma_{k-1}(G)}{2} + 2^{k-2}, \gamma_{k-1}(G)\right), \gamma_{k-2}(G)\right).$$

We can slightly improve the pebbling result if we do some case analysis.

Theorem 7 *For all $k \geq 3$ and any graph G whose edge set is non empty we have:*

$$\pi_{\text{opt}}(G) \geq \min\left(2^k, \gamma_{k-1}(G) + 2^{k-2} + 1, \gamma_{k-2}(G) + 1\right).$$

PROOF: The previous proof immediately gives the desired result if $|\operatorname{supp}(P)| \neq \gamma_{k-1}(G)$ or one of the inequalities in $2^{2-k}(|P| - |\operatorname{supp}(P)|) \geq W_{P\tau|N(\operatorname{supp}(P))}(v) \geq 1$ is strict. Therefore we investigate the case when $|\operatorname{supp}(P)| = \gamma_{k-1}(G)$ and show that one of the inequalities is strict. We use again the assumption that $|P| < \min(\gamma_{k-2}(G) + 1, 2^k)$.

Suppose that $\gamma_{k-1}(G) = 1$. Then P contains pebbles only at a vertex u. Since $\operatorname{supp}(P)$ is still not a distance $k-2$ domination set, there is a vertex v whose distance from u is $k-1$. Thus 2^{k-1} pebbles at u are required to reach v and these are also enough. So $\pi_{\text{opt}}(G) = 2^{k-1} \geq \gamma_{k-1}(G) + 2^{k-2} + 1$.

Otherwise $\gamma_{k-1}(G) \geq 2$. Therefore for each $p \in \operatorname{supp}(P)$ there is a vertex v, such that the distance between v and p is $k-1$ but the distance between v and $\operatorname{supp}(P) \setminus \{p\}$ is at least k.

Fix p and v and choose a σ sequence of pebbling moves which moves a pebble to v and divide it to τ and μ like in the previous proof.

If τ removes more than two pebbles from a vertex, then at least two pebbles are consumed there and we have counted at most one consumption at each vertex, hence $|P| - |\operatorname{supp}(P)| > \left|P_\tau|_{N(\operatorname{supp}(P))}\right|$ and the first inequality is strict.

If τ contains a pebbling move which removes two pebbles from a $q \in \operatorname{supp}(P) \setminus \{p\}$, then this move places a pebble to a vertex u whose distance from v is $k-1$. If another move does not move forward this pebble, then $P_{\tau|N(\operatorname{supp}(P)}(u) > 0$ and its coefficient in $W_{P\tau|N(\operatorname{supp}(P))}(v)$ is at most 2^{1-k} which is smaller than 2^{2-k} and the first equality is not possible. Else, a pebbling move removes two pebbles from u and consumes a pebble. We have not counted this consumption, hence $|P| - |\operatorname{supp}(P)| > |P\tau|N(\operatorname{supp}(P))|$.

The only remaining case is when τ contains only one pebbling move which moves a pebble from p to a vertex w. μ can use only this pebble, but one pebble is not enough to apply a single pebbling move, therefore μ does nothing, w is not v because the distance between them is at least two, so σ does not move a pebble to v which is a contradiction. Therefore this case is not possible. □

Using this last version of our result we can determine the optimal pebbling number of $K_3 \square K_3 \square K_5$.

Corollary 8 *The optimal pebbling number of $K_3 \square K_3 \square K_5$ is 6.*

PROOF: We have already seen a solvable distribution with size six in Figure 1. It is not hard to see that the distance 2 domination number of $K_3 \square K_3 \square K_5$ is three:

The support of the given distribution is a distance 2 domination set on three vertices. Two vertices are not enough. Consider a set S whose size is two. The graph is vertex transitive, therefore it does not matter how we chose the first vertex s_1. In each copy of $K_3 \square K_3$ which does not contain s_1 there are four undominated vertices whose distance from s_1 is more than two. The intersection of the closed neigbourhoods of the undominated vertices which are contained in the same $K_3 \square K_3$ is empty. After we chose s_2 there will be a $K_3 \square K_3$ which contains neither s_1 nor s_2. To reach its undominated vertices we have to move to a different $K_3 \square K_3$, which consumes one of the two moves but our location in $K_3 \square K_3$ does not change during this move. Only one more remained in $K_3 \square K_3$, but this is not enough to arrive all four undominated vertices, because the intersection of their closed neighbourhoods is empty. Therefore S is not a distance 2 domination set.

The domination number of $K_3 \square K_3 \square K_5$ is more than 4:

Consider a set of vertices S whose size is 4. The pigeonhole principle implies that there is a $K_3 \square K_3$ which does not contain an element of S. Two vertex from the same $K_3 \square K_3$ have some common adjacent vertices but all of them are contained in the same $K_3 \square K_3$ where the two original vertices. Therefore

each vertex in this $K_3 \square K_3$ requires a different element of S which dominates it. The order of $K_3 \square K_3$ is nine, therefore S is not a domination set.

Finally we set k to 3 and apply Theorem 7. $\square$

References

[1] C. Belford, N. Sieben Rubbling and optimal rubbling of graphs, *Discrete Mathematics* **309** (2009) pp. 3436–3446.

[2] D.P. Bunde, E. W. Chambers, D. Cranston, K. Milans, D. B. West, Pebbling and optimal pebbling in graphs *J. Graph Theory* **57 no. 3.** (2008) pp. 215–238.

[3] F. Chung, Pebbling in hypercubes, *SIAM J. Discrete Math.* **2** (1989) pp. 467–472.

[4] H. Fu, C. Shiue, The optimal pebbling number of the caterpillar, *Taiwanese Journal of Mathematics* **13 no. 2A** (2009) pp. 419–429.

[5] H. Fu, C. Shiue, The optimal pebbling number of the complete m-ary tree, *Discrete Mathematics* **222 no. 1–3** (2000) pp. 89–100.

[6] T. Friedman, C.Wyels, Optimal pebbling of paths and cycles *arXiv:math/0506076 [math.CO]*

[7] E. Győri, G. Y. Katona, L. F. Papp, Constructions for the optimal pebbling of grids, *arXiv:1601.02229 [math.CO]*

[8] E. Győri, G. Y. Katona, L. F. Papp, C. Tompkins Optimal Pebbling Number of Staircase Graphs, *arXiv:1611.09686 [math.CO]*

[9] D. S. Herscovici, B. D. Gester, G. H. Hurlbert, Optimal pebbling in product of graphs *Australasian J. of Combinatorics* **50** (2011) pp. 3–24.

[10] G. Y. Katona, N. Sieben Bounds on the Rubbling and Optimal Rubbling Numbers of Graphs, *Electronic Notes in Discrete Mathematics* **38** (2011) pp. 487–492.

[11] G. Y. Katona, L. F. Papp The optimal rubbling number of ladders, prisms and Möbius-ladders, *Discrete Applied Mathematics* **209** (2016) pp. 227–246.

[12] K. Milans, B. Clark, The complexity of graph pebbling, *SIAM J. Discrete Mathematics* **20 no. 3** (2006) pp. 769–798.

[13] J. Muntz, S. Narayan, N. Streib, K. V. Ochten, Optimal pebbling of graphs, *Discrete Mathematics* **307** (2007) pp. 2315–2321.

[14] L. Pachter, H.S. Snevily, B. Voxman On pebbling graphs, *Congressus Numerantium* **107** (1995) pp. 65–80.

Counting Minimum Weight Arborescences

Koyo Hayashi

Department of Mathematical Informatics
University of Tokyo
Tokyo 113-8656, Japan
koyo_hayashi@mist.i.u-tokyo.ac.jp

Satoru Iwata[1]

Department of Mathematical Informatics
University of Tokyo
Tokyo 113-8656, Japan
iwata@mist.i.u-tokyo.ac.jp

Abstract: In a directed graph $D = (V, A)$ with a specified vertex $r \in V$, an arc subset $B \subseteq A$ is called an *r-arborescence* if B has no arc entering r and there is a unique path from r to v in (V, B) for each $v \in V \backslash \{r\}$. The problem for finding a minimum weight r-arborescence in a weighted digraph has been studied for decades starting with Chu and Liu (1965), Edmonds (1967) and Bock (1971). In this paper, we focus on the number of minimum weight arborescences. We present an algorithm for counting minimum weight r-arborescences in $O(n^\omega)$ time, where n is the number of vertices of an input digraph and ω is the matrix multiplication exponent.

Keywords: minimum weight arborescence, matrix tree theorem, counting

1 Introduction

In a directed graph $D = (V, A)$ with a specified vertex $r \in V$, an arc subset $B \subseteq A$ is called an *r-arborescence* (or an *arborescence rooted at r*) if B has no arc entering r and there is a unique path from r to v in (V, B) for each $v \in V \backslash \{r\}$. As easily checked, a digraph D contains an r-arborescence if and only if each vertex in D is reachable from r. If D is a weighted digraph, a *minimum weight r-arborescence* is an r-arborescence whose total arc weight is minimum. Polynomial-time algorithms for finding a minimum weight r-arborescence were discovered independently by Chu and Liu [3], Edmonds [4] and Bock [1]. The best known bound for this problem has been obtained by Gabow et al. [6]. Their algorithm runs in $O(m+n \log n)$ time, where n and m are the numbers of vertices and arcs of an input digraph, respectively.

The above algorithms, however, find at most one minimum weight r-arborescence, while a digraph might contain more than one. In this paper, we focus on the multiplicity of optimal solutions. More specifically, we consider the following problem:

$$\text{Given a directed graph } D = (V, A) \text{ with a specified vertex } r \in V \text{ and a weight function } w : A \to \mathbb{R}_+, \text{ find the number of minimum weight } r\text{-arborescences in } D, \tag{1}$$

where $\mathbb{R}_+$ is the set of nonnegative real numbers. If w is a uniform weight, in particular, this problem is easy. All we have to do in this case is to compute the number of r-arborecences in D. This can be done by applying the following theorem, which is commonly known as the Matrix Tree Theorem. See, e.g., [7, Problem 4.16] for its proof.

Theorem 1 (Matrix Tree Theorem) *Let $D = (V, A)$ be a directed graph. Let a_{ij} denote the number of arcs leaving i and entering j for any two distinct vertices $i, j \in V$. Define the $V \times V$ matrix $L = (l_{ij})$ by*

$$l_{ij} := \begin{cases} \sum_{k \neq j} a_{kj} & (i = j), \\ -a_{ij} & (\textit{otherwise}). \end{cases} \tag{2}$$

[1]Research is supported by CREST, JST.

Then, for each vertex $i \in V$, the number of arborescences in D rooted at i is equal to $\det L_i$, *where L_i is the submatrix obtained by deleting the i-th row and column from L.* $\square$

By Theorem 1, one can compute the number of minimum weight r-arborescences in a uniformly weighted digraph in $O(n^\omega)$ time, where ω is the matrix multiplication exponent ($2 < \omega < 3$), i.e., the number of elementary operations needed to multiply two $n \times n$ matrices is $O(n^\omega)$. Although problem (1) is not so simple for an arbitrary w, we show that one can solve it within the same asymptotic running time based on the method of Fulkerson [5].

The problem for finding a minimum weight r-arborescence can be formulated as an integer program, which can be relaxed to the following linear program:

$$\begin{aligned} \text{Minimize} \quad & \sum_{a \in A} w(a)x(a) \\ \text{subject to} \quad & \sum_{a \in \delta^-(U)} x(a) \geq 1 \qquad (U \subseteq V \backslash \{r\}), \\ & x(a) \geq 0 \qquad (a \in A), \end{aligned} \tag{LP}$$

where $\delta^-(U)$ denotes the set of arcs entering U. The dual of (LP) can be described as follows:

$$\begin{aligned} \text{Maximize} \quad & \sum_{U \subseteq V \backslash \{r\}} y(U) \\ \text{subject to} \quad & \sum_{\substack{U \subseteq V \backslash \{r\}: \\ a \in \delta^-(U)}} y(U) \leq w(a) \qquad (a \in A), \\ & y(U) \geq 0 \qquad (U \subseteq V \backslash \{r\}). \end{aligned} \tag{DP}$$

Fulkerson [5] gave an algorithm for finding an optimal solution of (DP). This algorithm yields as a byproduct an arc subset $A^\circ \subseteq A$ and a collection $\mathcal{F} \subseteq 2^{V \backslash \{r\}}$ such that an r-arborescence B in D is of minimum weight if and only if

$$B \subseteq A^\circ \text{ and } |B \cap \delta^-(U)| = 1 \text{ for each } U \in \mathcal{F}. \tag{3}$$

This condition comes from the complementary slackness between (LP) and (DP).

To solve our problem (1), it suffices to count r-arborescences that satisfy (3). Actually, such counting can be done in $O(n^\omega)$ time. A key observation that leads to this bound is that:

> Given an unweighted strongly connected digraph $D = (V, A)$ with n vertices, one can determine the numbers of arborescences in D rooted at v for all $v \in V$ simultaneously in $O(n^\omega)$ time. (4)

We also give an efficient implementation of Fulkerson's algorithm that runs in $O(n^2 + m \log n)$ time, which enables us to solve our problem (1) in $O(n^\omega)$ time as a whole.

A similar problem for spanning trees in an undirected graph has already been considered by Broder and Mayr [2]. They devised an algorithm for counting minimum weight spanning trees in an undirected graph in $O(n^\omega)$ time.

The rest of this paper is organized as follows. In Section 2, we explain Fulkerson's algorithm. In Section 3, we give an $O(n^2 + m \log n)$-time implementation of Fulkerson's algorithm. Section 4 is devoted to proving (4). In Section 5, we give an algorithm for counting r-arborescences that satisfy (3) in $O(n^\omega)$ time to conclude that one can solve our problem (1) in $O(n^\omega)$ time.

2 Fulkerson's algorithm

Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$ and let $w : A \to \mathbb{R}_+$ be a weight function. We assume that every vertex in D is reachable from r and that no arc of A enters r. In this section, we

explain Fulkerson's algorithm [5] for finding an optimal solution y of (DP). An arc $a \in A$ is said to be *tight* for a feasible solution y of (DP) if

$$\sum_{U \subseteq V\setminus\{r\}:\, a \in \delta^-(U)} y(U) = w(a). \tag{5}$$

Fulkerson's algorithm can be described as follows.

ALGORITHM FULKERSON

⟨**1**⟩ Set $A^\circ := \{a \in A \mid w(a) = 0\}$ and $y := 0$.
⟨**2**⟩ Iterate the following until every vertex is reachable in $D^\circ = (V, A^\circ)$ from r.
  ⟨**2-1**⟩ Find a strong component U of the digraph D° with $r \notin U$ and $A^\circ \cap \delta^-(U) = \emptyset$.
  ⟨**2-2**⟩ Increase $y(U)$ as much as possible until some arc $a \in \delta^-(U)$ gets tight for y.
  ⟨**2-3**⟩ Set $A^\circ := A^\circ \cup \{a \in \delta^-(U) \mid a \text{ is tight for } y\}$.

Let y and A° be those obtained at the end of the above algorithm and set $D^\circ := (V, A^\circ)$. Note that an arc a belongs to A° if and only if a is tight for y. Let $\mathcal{F}$ be a collection of vertex sets $U \subseteq V\setminus\{r\}$ with $y(U) > 0$.

For a vertex subset $U \subseteq V$, we denote by $D^\circ[U]$ the subgraph of D° induced by U. It is easy to see that $D^\circ[U]$ is strongly connected for each $U \in \mathcal{F}$, and that $\mathcal{F}$ is laminar, i.e., $U \subseteq W$, $W \subseteq U$ or $U \cap W = \emptyset$ for all $U, W \in \mathcal{F}$. Then we can take an r-arborescence B in D such that $B \subseteq A^\circ$ and $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$, by taking $D = D^\circ$ in the following lemma.

Lemma 2 *Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$ and let $\mathcal{F} \subseteq 2^{V\setminus\{r\}}$ be a laminar family. Suppose that every vertex is reachable in D from r and $D[U]$ is strongly connected for each $U \in \mathcal{F}$. Then there exists an r-arborescence B in D such that $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$.*

PROOF: By induction on $|\mathcal{F}|$. We may assume that $\mathcal{F}$ contains no singleton, since any r-arborescence in D enters each vertex $v \neq r$ exactly once. The case $\mathcal{F} = \emptyset$ being trivial, suppose that $|\mathcal{F}| \geq 1$. Let U be an inclusion-wise minimal set in $\mathcal{F}$. Shrink U to a single vertex u, obtaining a new digraph D'. Similarly, set $\mathcal{F}' := \{(W\setminus U) \cup \{u\} \mid U \subsetneq W \in \mathcal{F}\} \cup \{W \mid W \cap U = \emptyset, W \in \mathcal{F}\}$. Since $|\mathcal{F}'| = |\mathcal{F}| - 1$, induction gives an r-arborescence B' in D' such that $|B' \cap \delta^-_{D'}(W)| = 1$ for each $W \in \mathcal{F}'$. Expanding u to the original vertex set U, extend B' to an r-arborescence B in D (such an r-arborescence exists since $D[U]$ is strongly connected). Then B satisfies that $|B \cap \delta^-_D(W)| = 1$ for each $W \in \mathcal{F}$. □

Choose any r-arborescence B in D such that $B \subseteq A^\circ$ and $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$. Now let us show that y is an optimal solution of (DP) and B is a minimum weight r-arborescence. To prove this, let χ^B be the incidence vector of B: namely we let $\chi^B(a)$ to be 1 for $a \in B$ and 0 otherwise. Then χ^B and y are optimal solutions of (LP) and (DP), respectively, by the complementary slackness. Indeed, if $\chi^B(a) > 0$ for some $a \in A$, then a is tight for y (as $a \in B \subseteq A^\circ$); If $y(U) > 0$ for some $U \subseteq V\setminus\{r\}$, then we have $\chi^B(\delta^-(U)) = 1$ (as $U \in \mathcal{F}$ and $|B \cap \delta^-(U)| = 1$). Hence y is optimal and B is of minimum weight.

Conversely, any minimum weight r-arborescence B in D satisfies that $B \subseteq A^\circ$ and $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$. To see this, let B^* be a minimum weight r-arborescence in D and let χ^{B^*} be the incidence vector of B^*. Then χ^{B^*} is an optimal solution of (LP). Since y is optimal as well, we have the following by the complementary slackness: For each $a \in B^*$, a is tight for y (as $\chi^{B^*}(a) > 0$); For each $U \in \mathcal{F}$, $\chi^{B^*}(\delta^-(U))$ is equal to 1 (as $y(U) > 0$). Hence we have $B^* \subseteq A^\circ$ and $|B^* \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$.

As a consequence we have the following well-known proposition.

Proposition 3 *Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$ such that every vertex in D is reachable from r, and let $w : A \to \mathbb{R}_+$ be a weight function. Then there exists an arc set $A^\circ \subseteq A$ and a laminar family $\mathcal{F} \subseteq 2^{V\setminus\{r\}}$ such that an r-arorescence B in D is of minimum weight if and only if $B \subseteq A^\circ$ and $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$.* □

3 An efficient implementation of Fulkerson's algorithm

In this section, we give an implementation of Fulkerson's algorithm that runs in $O(n^2 + m \log n)$ time. Fulkerson's algorithm can be redescribed as follows.

ALGORITHM FULKERSON*

⟨**1**⟩ Set $A^\circ := \emptyset$ and $y := 0$.
Set $z(v) := 0$ for each $v \in V \backslash \{r\}$.
⟨**2**⟩ Iterate the following until every vertex is reachable in $D^\circ = (V, A^\circ)$ from r.
⟨**2-1**⟩ Find a strong component U of the digraph D° with $r \notin U$ and $A^\circ \cap \delta^-(U) = \emptyset$.
⟨**2-2**⟩ Set $\mu := \min\{w(a) - z(v) \mid a = (u, v) \in \delta^-(U)\}$.
Set $y(U) := \mu$.
Set $z(v) := z(v) + \mu$ for each $v \in U$.
Set $A^\circ := A^\circ \cup \{a \in \delta^-(U) \mid w(a) - z(v) = 0\}$.

Throughout the iterations, for each $v \in V \backslash \{r\}$, $z(v)$ is equal to the sum of all positive $y(W)$ with $v \in W \subseteq V \backslash \{r\}$. Note that the case $\mu = 0$ may occur in ⟨**2-2**⟩ if U is a singleton (as we set $A^\circ := \emptyset$ initially).

Let us consider the running time bound. First note that there are at most $2n$ iterations. This comes from the fact that the collection of vertex sets U chosen in ⟨**2-1**⟩ is laminar.

Next we consider the running time of ⟨**2-2**⟩. This part can be done in $O(n^2 + m)$ time throughout all iterations, if we initially sort arcs of $\delta^-(v)$ for each $v \in V \backslash \{r\}$ so that they are in increasing order with respect to w. (Indeed, A° never decreases throughout the iterations.) This sorting can be done in $O(m \log n)$ time. Hence we can perform ⟨**2-1**⟩ (including sorting) in $O(n^2 + m \log n)$ time throughout all iterations.

Now let us consider the running time of ⟨**2-1**⟩. A naive way for this part could require $O(m+n)$ time at each iteration, which amounts to $O(nm)$ time as a whole. Actually, one can do ⟨**2-1**⟩ in $O(n)$ time at each iteration. To show this, we introduce a certain concept for reachability.

Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$. We say that D is *r-harmonious* if there exist functions $\phi : V \to \{0, 1, 2\}$ and $\theta : V \to \mathbb{Z}_+$ such that:

(i) a vertex $v \in V$ is reachable from r if and only if $\phi(v) = 0$;
(ii) if $\phi(v) = 2$, then $\delta^-(v) = \emptyset$;
(iii) for any $u, v \in V$ with $\phi(u) \geq 1$ and $\phi(v) = 1$, v is reachable from u if and only if $\theta(u) \geq \theta(v)$;
(iv) if D contains no r-arborescence, then there exists a strong component K of D such that $\delta^-(K) = \emptyset$ and $\phi(v) = 1$ for all $v \in K$. (6)

If such functions ϕ and θ exist, we say that ϕ is a *color* for D and θ is a *label* for D. See Figure 1 for an example of an r-harmonious digraph.

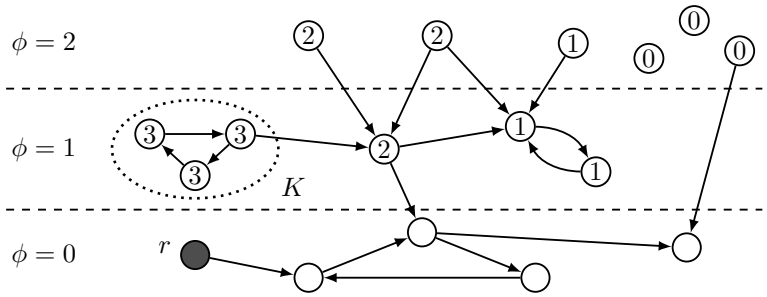


Figure 1: An r-harmonious digraph. Numbers assigned to vertices represent their values of the label.

Let us check some facts on an r-harmonious digraph $D = (V, A)$ with a color ϕ and a label θ. Suppose that D contains no r-arborescence, and let K be a strong component of D such that $\delta^-(K) = \emptyset$ and $\phi(v) = 1$ for all $v \in K$. Then each vertex v with $\phi(v) = 1$ is reachable from any vertex of K. (Indeed, if $\theta(u) < \theta(v)$ for some $u \in K$ and some vertex v with $\phi(v) = 1$, then u is reachable from v but v is not reachable from u, which contradicts that K is a strong component of D with $\delta^-(K) = \emptyset$.) This also implies the uniqueness of K. So we let $K(D)$ denote the strong component K of D for any r-harmonious digraph D that contains no r-arborescence.

Now we are ready to prove the following lemma, which will be used to implement $\langle$**2-1**$\rangle$ efficiently.

Lemma 4 *Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$. Suppose that D is r-harmonious with a color ϕ and a label θ and that D contains no r-arborescence. When adding to D some arcs entering $K(D)$, one can find a color and a label for the new digraph in $O(n)$ time. In particular, the new digraph is r-harmonious.*

Proof: Let F be the set of arcs entering $K(D)$ that have been added to D, and set $T := \{u \mid (u, v) \in F\}$. Let $D' := (V, A \cup F)$ be the new digraph. Clearly, (6)(ii) is maintained. We consider two cases.

Case 1: $T \cap \phi^{-1}(0) \neq \emptyset$. In this case, every vertex v with $\phi(v) = 1$ becomes reachable in D' from r (as $K(D)$ is reachable in D' from r). Set $\phi(v) := 0$ for each vertex v with $\phi(v) = 1$, and set $\theta(v) := 0$ for all $v \in V$. This maintains condition (6)(i), (ii) and (iii). If D' contains no r-arborescence, choose a vertex s with $\phi(s) = 2$, and set $\phi(s) := 1$ and $\theta(s) := 1$. This maintains (6). (In fact, $K(D') = \{s\}$.)

Case 2: $T \cap \phi^{-1}(0) = \emptyset$. Clearly, (6)(i) is maintained. Let k be the value of θ on $K(D)$. (Note that θ takes the same value over $K(D)$.) Let j be the minimum value of θ over $K(D) \cup \{v \in T \mid \phi(v) = 1\}$. To restore (6)(iii), we do the following:

$$\text{Set } \theta(v) := j \text{ for all } v \in \{u \in V \mid \phi(u) \geq 1,\ j \leq \theta(u) \leq k\} \cup \{u \in T \mid \phi(u) = 2\}. \tag{7}$$

This maintains (6)(iii), since the set of vertices u with $j \leq \theta(u) \leq k$ and $\phi(u) = 1$ is a strong component of D'.

If $\theta(v)$ is less than j for any vertex v with $\phi(v) = 2$ after doing (7), then (6)(iv) is also maintained. (In fact, $K(D') = \{u \in V \mid \phi(u) = 1, \theta(u) = j\}$.) If there is a vertex s with $\phi(s) = 2$ and $\theta(s) = j$ after doing (7), we set $\phi(s) := 1$ and $\theta(s) := j + 1$. This maintains (6). (In fact, $K(D') = \{s\}$.)

It is not difficult to see that these operations can be done in $O(n)$ time. □

Lemma 4 implies that $\langle$**2-1**$\rangle$ can be performed efficiently by keeping $D^\circ = (V, A^\circ)$ r-harmonious throughout the iterations. So we have the following result.

Theorem 5 *Fulkerson's algorithm can be implemented to run in $O(n^2 + m \log n)$ time.*

Proof: Since we have already observed that $\langle$**2-2**$\rangle$ can be done in $O(n^2 + m \log n)$ time throughout all iterations, it suffices to show that one can perform $\langle$**2-1**$\rangle$ in $O(n)$ time at each iteration.

At the start of the algorithm, do the following: Choose a vertex $s \in V \backslash \{r\}$ arbitrarily, and set $\phi(r) := 0$, $\phi(s) := 1$ and $\phi(v) := 2$ for all $v \in V \backslash \{r, s\}$; Set $\theta(s) := 1$ and set $\theta(v) := 0$ for all $v \in V \backslash \{s\}$. Then $D^\circ = (V, A^\circ)$ is initially an r-harmonious digraph with the color ϕ and the label θ (as $A^\circ = \emptyset$). If we choose $K(D^\circ)$ in $\langle$**2-1**$\rangle$ at each iteration, we can restore ϕ and θ in $O(n)$ time for the new digraph D° obtained after $\langle$**2-2**$\rangle$ at its iteration by Lemma 4. This implies that one can perform $\langle$**2-1**$\rangle$ in $O(n)$ time at each iteration. □

Directly from Theorem 5 and discussion in Section 2, we have the following corollary.

Corollary 6 *Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$ such that every vertex in D is reachable from r, and let $w : A \to \mathbb{R}_+$ be a weight function. Then one can find in $O(n^2 + m \log n)$ time an arc set $A^\circ \subseteq A$ and a laminar family $\mathcal{F} \subseteq 2^{V \backslash \{r\}}$ such that an r-arorescence B in D is of minimum weight if and only if $B \subseteq A^\circ$ and $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$.* □

4 Simultaneous counting of rooted arborescences

Let $D = (V, A)$ be an unweighted digraph with n vertices. For each $v \in V$, let $\tau(v)$ denote the number of arborescences in D rooted at v. Theorem 1 tells us that one can find $\tau(v)$ in $O(n^\omega)$ time for each vertex v. To find $\tau(v)$ for all $v \in V$, we must determine all the diagonal cofactors of the matrix L defined by (2). If L were nonsingular, this could be done by computing its inverse in $O(n^\omega)$ time. Unfortunately, however, L is in fact singular, and a naive way requires $O(n^\omega \cdot n)$ time. In this section, we show that one can determine $\tau(v)$ for all $v \in V$ in $O(n^\omega)$ time if D is strongly connected.

Let $H = (h_{ij})$ be an $n \times n$ matrix satisfying the following condition:

$$\sum_{i=1}^{n} h_{ij} = 0 \quad \text{for } j = 1, 2, \ldots, n. \tag{8}$$

So the sum of each column of H is equal to zero. Let H_i denote the submatrix obtained by deleting the i-th row and column from H for $i = 1, 2, \ldots, n$. Then H can be partitioned as

$$H = \begin{pmatrix} \alpha & \beta \\ \gamma & \eta \end{pmatrix}, \tag{9}$$

where $\alpha = H_n$. If α is nonsingular, then each $\det H_i$ can be written as follows.

Lemma 7 *If H satisfies (8) and $\alpha = H_n$ is nonsingular, then*

$$\det H_i = -(\alpha^{-1}\beta)_i \cdot \det \alpha \tag{10}$$

for $i = 1, 2, \ldots n-1$. Here $(\alpha^{-1}\beta)_i$ means the i-th component of the vector $\alpha^{-1}\beta$.

Proof: Let i be an integer from 1 to $n-1$. Define

$$P := \begin{pmatrix} \alpha^{-1} & 0 \\ \lambda & 1 \end{pmatrix}, \tag{11}$$

where λ is a row vector of dimension $n-1$ with all entries equal to 1. Then we have

$$PH = \begin{pmatrix} I & \alpha^{-1}\beta \\ 0 & 0 \end{pmatrix}, \tag{12}$$

where I is the identity matrix of dimension $n-1$. Let e_i denote the i-th unit vector, and let G_i denote the matrix arising from H by replacing the i-th column of H with e_i. It follows from (12) that

$$\det(PG_i) = \det \begin{pmatrix} (Pe_i)_i & (\alpha^{-1}\beta)_i \\ (Pe_i)_n & 0 \end{pmatrix} = -(\alpha^{-1}\beta)_i. \tag{13}$$

This implies the lemma, since $\det(PG_i) = \det P \det G_i = \det H_i / \det \alpha$. □

Since both $\det \alpha$ and the vector $\alpha^{-1}\beta$ can be determined in $O(n^\omega)$ time, Lemma 7 implies that all the diagonal cofactors of the matrix H can be computed in $O(n^\omega)$ time. This immediately yields the following result.

Theorem 8 *Given a strongly connected digraph $D = (V, A)$, one can determine $\tau(v)$ for all $v \in V$ in $O(n^\omega)$ time, where $\tau(v)$ is the number of arborescences in D rooted at v.*

Proof: Define the matrix L by (2). Note that the sum of each column of L is equal to zero. Since D is strongly connected, $\det L_i$ is positive for each $i \in V$ by Theorem 1. This implies, in particular, that L_i is nonsingular for any $i \in V$. Hence, directly from Lemma 7 and Theorem 1, we obtain the theorem. □

5 Counting minimum weight arborescences

Proposition 3 reduces our problem (1) to a problem for counting arborescences satisfying certain conditions in an unweighted digraph. Recall that $D[U]$ denotes the subgraph of D induced by U. We now consider the following problem:

$$\begin{aligned} &\text{Given: a digraph } D=(V,A) \text{ with a specified vertex } r\in V \text{ and a laminar family } \mathcal{F}\subseteq 2^{V\setminus\{r\}} \\ &\qquad \text{such that every vertex is reachable in } D \text{ from } r \text{ and } D[U] \text{ is strongly connected for} \\ &\qquad \text{each } U\in\mathcal{F}; \\ &\text{Find: the number of } r\text{-arborescences } B \text{ in } D \text{ such that } |B\cap\delta^-(U)|=1 \text{ for each } U\in\mathcal{F}. \end{aligned} \tag{14}$$

In this section, we show that one can solve the above problem (14) in $O(n^\omega)$ time. With Corollary 6, this implies that our problem (1) can be solved in $O(n^\omega)$ time.

5.1 Outline

We may assume that $\mathcal{F}$ contains no singleton, since any r-arborescence in D enters each vertex $v\neq r$ exactly once. We say that an r-arborescence B in D is *$\mathcal{F}$-tight* if $|B\cap\delta^-(U)|=1$ for each $U\in\mathcal{F}$. A key observation for counting $\mathcal{F}$-tight r-arborescences in D is that:

$$\text{For any } \mathcal{F}\text{-tight } r\text{-arborescence } B \text{ in } D \text{ and for any } U\in\mathcal{F},\ B[U] \text{ is an arborescence in } D[U], \tag{15}$$

where $B[U]$ denotes the set of arcs of B spanned by U.

Now we give an useful idea that yields an efficient method for counting $\mathcal{F}$-tight r-arborescences in D. Let U be an inclusion-wise minimal set in $\mathcal{F}$. For each $v\in U$, let $\tau_U(v)$ denote the number of arborescences in $D[U]$ rooted at v. Since $D[U]$ is strongly connected, $\tau_U(v)$ is positive for each $v\in U$. Let D_U be a digraph arising from D by doing the following operations:

$$\text{For each arc } a=(s,v)\in\delta^-(U), \text{ replace } a \text{ by } \tau_U(v) \text{ parallel arcs; Shrink } U \text{ to a new vertex } u. \tag{16}$$

Similarly, let $\mathcal{F}_U$ be a collection obtained from $\mathcal{F}$ by shrinking U. More precisely, set $\mathcal{F}_U:=\{(W\setminus U)\cup\{u\}\,|\,U\subsetneq W\in\mathcal{F}\}\cup\{W\,|\,W\cap U=\emptyset,\ W\in\mathcal{F}\}$. Then we have the following.

Claim 9 *The number of $\mathcal{F}$-tight r-arborescences in D is equal to that of $\mathcal{F}_U$-tight r-arborescences in D_U.*

Proof: Let D' be a digraph obtained from D by shrinking U to one new vertex u (without replicating arcs). To avoid complication, let $\delta^-(U)$ and $\delta^-(u)$ denote the arc sets $\delta^-_D(U)$ and $\delta^-_{D'}(u)$, respectively, and identify them as the same set. For each $a\in\delta^-(U)$, let $\partial^- a$ denote the head of a in D. So $\partial^- a\in U$.

For each arc $a\in\delta^-(u)$, let $\sigma(a)$ denote the number of $\mathcal{F}_U$-tight r-arborescences in D' that contains a. Then for each arc $a\in\delta^-(U)$ the number of $\mathcal{F}$-tight r-arborescences in D that contains a is equal to $\sigma(a)\cdot\tau_U(\partial^- a)$, by (15) and the minimality of U. Hence the total number of $\mathcal{F}$-tight r-arborescences in D is equal to $\sum_{a\in\delta^-(U)}\sigma(a)\cdot\tau_U(\partial^- a)$, which implies the claim. □

We can derive from Claim 9 an efficient method for solving problem (14). Note that $|\mathcal{F}_U|=|\mathcal{F}|-1$. Resetting $D:=D_U$ and $\mathcal{F}:=\mathcal{F}_U$ and iterating the series of the operations, we will get $\mathcal{F}=\emptyset$ at some point. Claim 9 implies that throughout the iterations the number of $\mathcal{F}$-tight r-arborescences in D does not change. If $\mathcal{F}$ is empty, the number of $\mathcal{F}$-tight r-arborescences in D is nothing but that of r-arborescences in D, which can be determined by just applying Theorem 1.

5.2 Algorithm description and complexity

Now we describe an algorithm for problem (14). Let $D=(V,A)$ be a digraph with vertex set $V=\{1,2,\ldots,n\}$ that contains a specified vertex $r\in V$, and let $\mathcal{F}\subseteq 2^{V\setminus\{r\}}$ be a laminar family. Suppose that D contains an r-arborescence and that $D[U]$ is strongly connected for each $U\in\mathcal{F}$. We assume

that $\mathcal{F}$ contains no singleton. Moreover, we assume that the laminar family $\mathcal{F} = \{U_k\}_{k=1}^t$ satisfies that $U_i \subsetneq U_j$ or $U_i \cap U_j = \emptyset$ for any $1 \leq i < j \leq t$, since the members of $\mathcal{F}$ can be found in such an order by Fulkerson's algorithm. Let a_{ij} be the number of arcs leaving i and entering j for any two distinct vertices $i, j \in V$. Then the counting algorithm for problem (14) can be described as follows.

ALGORITHM COUNTING

⟨**1**⟩ Set $\psi(v) := v$ for each vertex $v \in V = \{1, 2, \ldots, n\}$. Set $q := n + 1$.
⟨**2**⟩ For $k = 1, 2, \ldots, t$, do the following.
⟨**2-1**⟩ Set $I := \{\psi(v) \mid v \in U_k\}$ and $J := \{\psi(v) \mid v \in V \backslash U_k\}$. Define the $I \times I$ matrix $L = (l_{ij})$ by

$$l_{ij} := \begin{cases} \sum_{p \in I \backslash \{i\}} a_{pj} & (i = j), \\ -a_{ij} & (\text{otherwise}). \end{cases} \tag{17}$$

Determine $\tau(i) := \det L_i$ for each $i \in I$, where L_i is the submatrix obtained by deleting i-th row and column from L.
⟨**2-2**⟩ Set $a_{jq} := \sum_{i \in I} a_{ji} \cdot \tau(i)$ and $a_{qj} := \sum_{i \in I} a_{ij}$ for each $j \in J$.
⟨**2-3**⟩ Set $\psi(v) := q$ for each $v \in U_k$. Set $q := q + 1$.
⟨**3**⟩ Set $I := \{\psi(v) \mid v \in V\}$. Define the $I \times I$ matrix $L = (l_{ij})$ by (17). Return $\det L_{\psi(r)}$.

Let us consider the running time bound. For $k = 1, 2, \ldots, t$, let d_k and n_k be the sizes of I and $I \cup J$ in ⟨**2-1**⟩ at k-th iteration, respectively. So $n_1 = n$. Note that each d_k is larger than 1 (since $\mathcal{F}$ contains no singleton). It is easy to see that $n_{k+1} = n_k - d_k + 1$ for $k = 1, 2, \ldots, t-1$. This gives that $t \leq n$ and $\sum_{k=1}^t d_k \leq 2n$. Since we can do ⟨**2-1**⟩ in $O({d_k}^\omega)$ time at k-th iteration by Theorem 8, we can perform ⟨**2-1**⟩, ⟨**2-2**⟩ and ⟨**2-3**⟩ at k-th iteration in time

$$O({d_k}^\omega + d_k(n_k - d_k) + n) \leq O({d_k}^\omega + nd_k). \tag{18}$$

Hence, throughout all iterations, we can perform ⟨**2**⟩ in time

$$O\left(\sum_{k=1}^t ({d_k}^\omega + nd_k)\right) \leq O\left(\left(\sum_{k=1}^t d_k\right)^\omega + n^2\right) \leq O(n^\omega). \tag{19}$$

Also we can do ⟨**3**⟩ in $O(n^\omega)$ time. Therefore, problem (14) can be solved in $O(n^\omega)$ time, which together with Corollary 6, implies the following theorem.

Theorem 10 *Given a directed graph $D = (V, A)$ with a specified vertex $r \in V$ and a weight function $w : A \to \mathbb{R}_+$, one can find the number of minimum weight r-arborescences in D in $O(n^\omega)$ time.* □

References

[1] F. BOCK: An algorithm to construct a minimum directed spanning tree in a directed network. In *Developments in Operations Research*, Gordon and Breach, New York, 1971, 29–44.

[2] A. Z. BRODER AND E. W. MAYR: Counting minimum weight spanning trees. *Journal of Algorithms*, **24** (1997), 171–176.

[3] Y. CHU AND T. LIU: On the shortest arborescence of a directed graph. *Scientia Sinica*, **14** (1965), 1396–1400.

[4] J. EDMONDS: Optimal branchings. *Journal of Research of the National Bureau of Standards*, **71B** (1967), 233–240.

[5] D. R. FULKERSON: Packing rooted directed cuts in a weighted directed graph. *Mathematical Programming*, **6** (1974), 1–13.

[6] H. N. Gabow, Z. Galil, T. Spencer and R. E. Tarjan: Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatrica*, **6** (1986), 109–122.

[7] L. Lovász: *Combinatorial Problems and Exercises: Second edition*, AMS Chelsea Publishing, Providence, 2007.

A Compact Representation for Modular Semilattices and Its Applications

Hiroshi Hirai

Graduate School of Information Science and Technology
The University of Tokyo
Tokyo 113-8656, Japan
hirai@mist.i.u-tokyo.ac.jp

So Nakashima

Graduate School of Information Science and Technology
The University of Tokyo
Tokyo 113-8656, Japan
so_nakashima@mist.i.u-tokyo.ac.jp

Abstract: A modular semilattice is a semilattice generalization of a modular lattice. We establish a Birkhoff-type representation theorem for modular semilattices, which says that every modular semilattice is isomorphic to the family of ideals in a certain poset with additional relations. This new poset structure, which we axiomize in this paper, is called a PPIP (projective poset with inconsistent pairs). A PPIP is a common generalization of a PIP (poset with inconsistent pairs) and a projective ordered space. The former was introduced by Barthélemy and Constantin for extablishing Birkhoff-type theorem for median semilattices, and the latter by Herrmann, Pickering, and Roddy for modular lattices. We show the $\Theta(n)$ respresentation complexity and a construction algorithm for PPIP-representations of $(\wedge, \vee)$-closed sets in the product L^n of modular semilattice L, which are also modular semilattices. This generelizes results of Hirai and Oki for a special median semlattice S_k. We also investigate implicational bases for modular semilattices. Extending results by Wild and Herrmann for modular lattices, we determine optimal implicational bases and develope a polynomial time recognition algorithm for modular semilattices. These results can be applied to retain the minimizer set of a submodular function on a modular semilattice.

Keywords: modular semilattice, Birkhoff representation theorem, implicational base, submodular function.

1 Introduction

The Birkhoff representation theorem says that every distributive lattice is isomorphic to the family of ideal in a poset (partially ordered set). This representation of a distributive lattice L is *compact* in the sence that the cardinarity of the poset is at most the height of L, and consequently has brought numerous algorithmic successes in discrete applied mathematics. The family of all stable matchings in the stable matching problem forms a distributive lattice, and is compactly represented by a poset. Several game-theoretic problems on stable matchings are elegantly solved by utilizing this poset repsensentation. The family of minimum s-t cuts in a network forms a distributive lattice. More generally, the family of minimizers of a *submodular set function* is a distributive lattice, and admits such a compact representation; see [9]. A canonical block-triangular form of a matrix by means of row and column permutations, known as the *Dulmage-Mendelsohn decomposition (DM-decomposition)*, is obtained via a maximal chain of the family of minimizers of a submodular function, in which a maximal chain corresponds to a topological order of the poset representation of the family. The DM-decomposition is further generalized to the *combinatorial canonical form (CCF)* of a *mixed matrix* [22, 23], which is also built on the same idea.

The present paper addresses Birkhoff-type compact representations for lattices and semilattices *beyond* distributive lattices. Here, by a compact representation of lattice or semilattice L we naively mean a structure whose size is smaller than the size of L and from which the original lattice structure can be recovered. Some of previous works relating this subject are explained as follows.

Median semilattices are a semilattice generalization of a distributive lattice, in which every principal ideal is a distributive lattice. Barthélemy and Constantin [5] established a Birkhoff-type reprsentation theorem for a median semilattice. Their theorem says that every median semilattice is compactly represented by, or more specifically, is isomorphic to the family of special ideals of a poset with an additional relation, called an *inconsistent relation*. This structure is called a *poset with inconsistent pairs (PIP)*, which was also independenly introduced by Nielsen, Plotkin, and Winskel [25] as a model of cocurrency in theoretical computer science, and recently rediscovered by Ardila, Owen, and Sullivant [2] from the state complex of robot motion planning; the name PIP is due to them. Hirai and Oki [16] applied PIP to represent the minimizer set of a *k-submodular function*, which is a generalization of a submodular set funtion defined on the product ${S_k}^n$ of a special median semilattice S_k (consisting of $k+1$ elements). They obtained several basic algorithmic results for this PIP-representation.

Modular lattices are a well-know lattice class that includes distributive lattices. Herrmann, Pickering, and Roddy [11] established a Birkhoff-type representation theorem of a modular lattice, which says that every modular lattice is isomorphic to the family of special ideals of a poset with an additional ternary relation, called a *collinear relation*. This structure is called a *projective ordered space*, and is viewed as a generalization of a *projective space*, which is a fundamental class of incidence geometries [27].

A theory of *implicational systems* (or *Horn formulas*) also provides a theoretical basis of compact reprsentations of lattice and semilattice; see recent survey [29]. Wild [28] determines an optimal implicational base (or a minimum-size Horn formula) of a modular lattice L, where L is regarded as a *closure system* $\mathcal{F} \subseteq 2^E$ on a suitable set E. This result is remarkable since obtaining an optimal implicational base is NP-hard in general. Subsequently, by utilizing the axiom of projective ordered space, Herrman and Wild [12] developed a polynomial time algorithm to decides whether a closure system given by implications is a modular lattice.

The goal of the paper is to generalize these results to a *modular semilattice*, which is a common generalization of a median semilattice and a modular lattice, and first appeared in a paper [4] of Bandelt, van de Vel, and Verheul. Recently, modular semilattices have unexpectedly emerged from several well-behaved classes of combinatorial optimization problems, and been being recognized as a next stage on which submodular function theory should be developed [14, 15]. The motivation of this paper comes from these emergences and future contribution of modular semilattices to combinatorial optimzation.

The results and the organization of this paper are outlined as follows:

Section 2: We establish a Birkhoff-type representation theorem for modular semilattices: Generalizing PIP and projective ordered space, we formulate the axiom of a new structure *PPIP* (projective poset with inconsistent relation), which is a certain poset endowed with both inconsistent and collinear relations. We prove a one-to-one correspondance between modular semilattices and PPIPs (Theorem 6). While projective ordered spaces generalize projective geometries, PPIP generalizes *polar spaces*, which are another fundamental class of incidence geometries.

Section 3: A typical emergence of a modular semilattice is as a $(\vee, \wedge)$-closed set B in the product L^n of a (very small) modular semilattice L. We investigate the representation complexity of such a modular semilattice B. We show that the number of $\vee$-irreducible elements of B is bounded by n times of the number of $\vee$-irreducible elements of L (Theorem 8). This attains a lower limit by Berman et al.[6], and in turn implies that the PPIP-representation for B is actually compact (i.e., has a polynomial size in n) provided the size of L is fixed. We give a polynomial time algorithm to construct PPIP assuming a membership oracle of B (Theorem 11), which is applied to the minimizer set of a submodular function on L^n. These generalize results of Hirai and Oki [16] for case of $L = S_k$.

Section 4: Extending Wild's result, we determine an optimal implicational base of a modular semilattice viewed as a $\cup$-closed family (Theorem 13). Utilizing the axiom of PPIP, we develop a polynomial time recognition algorithm for moduler semilattices given by implications (Theorem 15), which is also an extension of the algorithm by Herrman and Wild [12] for modular lattices.

These results have potential applications to (i) the computation of the PPIP-representation of the minimizer set of a submodular function on a modular semilattice and (ii) a canonical block-trianguation to a partitioned matrix [19], which is a futher generalization of the DM-decomposition. Details are found in the full version of this paper.

Notation

We use a standard terminology on posets and lattices. Let P be a poset. A subset $X \subseteq P$ is called an *ideal* if $p \leq p'$ and $p' \in X$ implies $p \in X$. The *principal ideal* of x, denoted by I_x, is the ideal $\{p \in P \mid p \leq x\}$. In this paper, semilattices are $\wedge$-semilattices. We assume that any chain in a semilattice have finite length. Let L be a semilattice. Note that the join $x \vee y$ exists if and only if there is a common upper bound of x and y. A semilattice L is said to be *modular* [4] if every principal ideal is a modular lattice, and for every $x, y, z \in L$, the join $x \vee y \vee z$ exists provided $x \vee y$, $y \vee z$, and $z \vee x$ exist. A *Median semilattice* [26] is a modular semilattice each of whose principal ideal is distributive. We say that $l \in L$ is $\vee$-*irreducible* if $l = a \vee b$ means $l = a$ or $l = b$. For a semilattice L, let L^{ir} denote the family of $\vee$-irreducible elements of L, where L^{ir} is a poset with the partial order derived from L. We denote $\{1, 2, \ldots, n\}$ by $[n]$. The symbol $|A|$ designates the cardinality of a set A.

2 Birkhoff-type representation

In this section, we introduce a new structure PPIP and establish a Birkhoff-type representation for modular semilattices. We suppose that all semilattices have finite length throughout this section. We first quickly review previous Birkhoff-type representations. In Section 2.1, we explain PIP representation for median semilattices by Barthélemy and Constantin [5]. In Section 2.2, we explain projective ordered space representation for modular lattices by Herrmann, Pickering, and Roddy [11]. In Section 2.3, we axiomatize PPIPs as a common generalization of PIPs and projective ordered spaces, and establish a Birkhoff-type representation theorem for modular semilattices.

2.1 Median semilattice and PIP

In this section, we introduce PIPs and explain a Birkhoff-type representation theorem for median semilattices. A key tool for providing compact representation is a poset endowed with an additional relation.

Let P be a poset. A symmetric binary relation $\smile$ defined on P is called an *inconsistent* relation [5] if the following conditions are satisfied:

(IC1) there are no common upper bounds of p and q provided $p \smile q$;

(IC2) if $p \smile q$, $p \leq p'$, and $q \leq q'$, then $p' \smile q'$.

Definition 1 *A PIP is a posed endowed with an inconsistent relation.*

Let P be a PIP. An *inconsistent pair* is a pair $(x, y) \in P^2$ such that $x \smile y$. A subset $X \subseteq P$ is said to be *consistent* if X contains no inconsistent pairs. We denote the family of consistent ideals of P by $\mathcal{C}(P)$. Regard $\mathcal{C}(P)$ as a poset with respect to the inclusion order $\subseteq$.

We define a symmetric binary relation $\smile$ on L^{ir} by $x \smile y$ if and only if $x \vee y$ does not exist for any $x, y \in P$. It was shown that $\smile$ is indeed an inconsistent relation [5]. We refer $\smile$ as an induced inconsistent relation. For median semilattice L, let $\mathrm{PIP}(L)$ denote the PIP which consists of $\vee$-irreducible elements of L and endowed with the induced partial order and inconsistent relation.

The following theorem establishes Birkhoff-type representation for median semilattices.

Theorem 2 ([5]) (1) *Let L be a median semilattice. Then $\mathcal{C}(\mathrm{PIP}(L))$ is isomorphic to L.*

(2) *Let P be a PIP. Then $\mathrm{PIP}(\mathcal{C}(P))$ is isomorphic to P.*

2.2 Modular lattice and projective ordered space

In this section, we introduce projective ordered spaces and explain a Birkhoff-type representation theorem for modular lattices. As in the case of median semilattice, a key tool for providing compact representation is posets endowed with an additional relation. In addition, the axiomatization of projective ordered spaces is necessary to establish our Birkhoff-type representation theorem.

Let P be a poset. A symmetric ternary relation C defined on P is called a *collinear relation* [11] if the following conditions are satisfied:

(CT1) p, q, and r are pairwise incomparable provided $C(p,q,r)$ holds;

(CT2) if $C(p,q,r)$ holds, $p \leq w$, and $q \leq w$, then $r \leq w$.

An *ordered space* is a posed endowed with a collinear relation. A triple of elements $x, y, z \in P$ is *collinear* if $C(x,y,z)$ holds. A *collinear triple* is a triple $(x,y,z) \in P^3$ such that $C(p,q,r)$ holds.

Let P be an ordered space. An ideal $X \subseteq P$ is called a *subspace* if $p, q \in X$ and the collinearity of p, q, r implies $r \in X$. Let $\mathcal{S}(P)$ be the family of subspaces of P. Regard $\mathcal{S}(P)$ as a poset with respect to the inclusion order $\subseteq$.

Let L be a semilattice. We define a symmetric ternary relation C on L^{ir} by $C(x,y,z)$ holds if and only if x, y, and z are pairwise incomparable, $x \vee y$, $y \vee z$, and $z \vee x$ exist, and they are equal. It was shown that C is indeed a collinear relation [11]. We refer C as a collinear relation induced by L. For modular lattice L, let $\mathrm{PS}(L)$ denote the ordered space which consists of $\vee$-irreducible elements of L and endowed with the induced partial order and collinear relation.

Though every PIP corresponds to a median semilattice, not all ordered spaces represent modular lattices. To avoid this inconvenience, Hermann, Pickering, and Roddy [11] axiomatized projective ordered spaces:

Definition 3 *An ordered space P is said to be projective if the following axioms are satisfied:*

(Regularity) *For any collinear triple (p,q,r) and $r' \in P$ such that $r' \leq r$, $r' \not\leq p$, and $r' \not\leq q$, there exist $p' \leq p$ and $q' \leq q$ such that $C(p',q',r')$ holds.*

(Triangle) *If $C(a,c,p)$ and $C(b,c,q)$ are satisfied, then at least one of the following conditions holds:*

- *There exists $x \in P$ such that $C(a,b,x)$ and $C(p,q,x)$ hold, $\{a,b,c,p,q,x\}$ are pairwise incomparable, and there are no collinear triples in $\{a,b,c,p,q,x\}$ other than (a,c,p), (b,c,q), (a,b,x), (p,q,x), and their permutations;*
- *There is $a' \leq a$ such that $C(b,q,a')$ holds;*
- *$C(b,q,p)$ holds;*
- *There are $a' \leq a$ and $p' \leq p$ such that $C(q,a',p')$ holds;*
- *$q \leq a$ or $q \leq p$.*

The following theorem establishes a Birkhoff-type representation theorem for modular lattices.

Theorem 4 ([11]) (1) *Let L be a modular lattice. Then $\mathrm{PS}(L)$ is a projective ordered space. Furthermore, $\mathcal{C}(\mathrm{PS}(L))$ is isomorphic to L.*

(2) *Let P be a projective ordered space. Then $\mathcal{S}(P)$ is a modular lattice. Furthermore, $\mathrm{PS}(\mathcal{S}(P))$ is isomorphic to P.*

2.3 Modular semilattice and PPIP

In this section, we establish a Birkhoff-type representation theorem for modular semilattice by introducing a new structure PPIP. A PPIP is a common generalization of a PIP and projective ordered space. Modular semilattices can be regarded as that of median semilattices and modular lattices. Therefore we expect that modular semilattices are compactly represented by such structures. In this paper, we only deal with PPIPs which satisfy *finite length condition*: there are no infinite chains of consistent subspaces.

Definition 5 *Let P be a poset associated with an inconsistent relation $\smile$ and collinear relation C. We say that P is a PPIP if the following axioms are satisfied:*

(Regularity) *The same as in Definition 3.*

(weak Triangle) *Suppose that $C(a,c,p)$ and $C(b,c,q)$ hold and $\{a,b,c,p,q\}$ is consistent. Then at least one of the five conditions of Triangle axiom in Definition 3 holds.*

(Consistent-Collinearity) *For any collinear triple (p,q,r), the following conditions are satisfied:*

- **(CC1)** *the set $\{p,q,r\}$ is consistent;*
- **(CC2)** *for any $x \in P$, x is consistent with either at most one of (p,q,r) or all of them.*

For a modular semilattice L, let $\mathrm{P}(L)$ denote L^{ir} equipped with the induced inconsistent relation, and collinear relation. We will later prove that $\mathrm{P}(L)$ is a PPIP if L is a modular semilattice. For a PPIP P, let $\mathcal{CS}(P)$ be the family of consistent subspaces of PPIP P. Regard $\mathcal{CS}(P)$ as a poset with respect to the inclusion order $\subseteq$.

We establish Birkhoff-type representation theorem for modular semilattices as follows:

Theorem 6 (1) *Let L be a modular semilattice. Then $\mathrm{P}(L)$ is a PPIP. Furthermore, $\mathcal{CS}(\mathrm{P}(L))$ is isomorphic to L. An isomorphism $\phi\colon L \to \mathcal{CS}(\mathrm{P}(L))$ is given by $\phi(l) := \{p \in \mathrm{P}(L) \mid p \le l\}$. The inverse ψ is given by $\psi(I) := \bigvee_{x\in I} x$. Here $\psi(\emptyset) = \min L$.*

(2) *Let P be a PPIP. Then $\mathcal{CS}(P)$ is a modular semilattice Furthermore, $\mathrm{P}(\mathcal{CS}(P))$ is isomorphic to P.*

In particular, modular semilattices are compactly represented by a PPIP.

Example 7 *A modular semilattice, illustrated in Figure 1 (a), is represented by the PPIP in Figure 1 (b).*

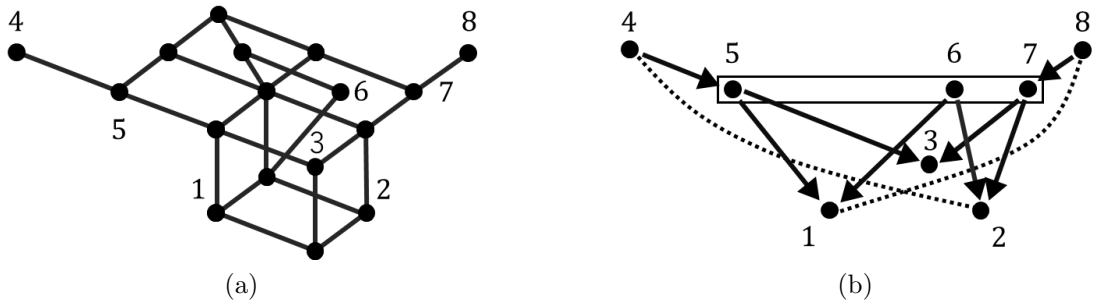


Figure 1: An example of PPIP representation. (a) Hasse diagram of a modular semilattice. Its $\vee$-irreducible elements are numbered. (b) PPIP representation of the modular semilattice. Dots and arrows constitute its Hasse diagram. Dotted line represents minimal inconsistent pairs (defined in Section 4.1). Three elements in the rectangular box are collinear.

3 $(\wedge, \vee)$-closed set in L^n

In this section, we deal with compact representations for $(\wedge, \vee)$-closed sets in L^n. In Section 3.1, we prove that any $(\wedge, \vee)$-closed set in L^n admits an $O(n)$-size representations. In Section 3.2, we address a polynomial time algorithm calculating the PPIP-representation of a $(\wedge, \vee)$-closed set in L^n. In this section, assume that all semilattices are finite.

3.1 $O(n|L^{ir}|)$-bound of $\vee$-irreducible elements

In this section, we deal with the upper bound of the size of compact representations for $(\wedge, \vee)$-closed sets in L^n. Let L be a semilattice. The symbol L^n denotes an n-product of L, whose partial order is the product order. Notice that we can calculate $\wedge$ and $\vee$ of L^n in the component-wise manner, that is, the following identity holds for any $\boldsymbol{l} = (l_1, l_2, \ldots, l_n) \in L^n$ and $\boldsymbol{l'} = (l'_1, l'_2, \ldots, l'_n)$:

$$\boldsymbol{l} \wedge \boldsymbol{l'} = (l_1 \wedge l'_1, l_2 \wedge l'_2, \ldots, l_n \wedge l'_n),$$
$$\boldsymbol{l} \vee \boldsymbol{l'} = (l_1 \vee l'_1, l_2 \vee l'_2, \ldots, l_n \vee l'_n) \ (\boldsymbol{l} \vee \boldsymbol{l'} \text{ exists if all } l_i \vee l'_i \text{ exist}).$$

A subset $B \subseteq L^n$ is said to be $(\wedge, \vee)$-*closed* if $b_1 \wedge b_2 \in B$ for any $b_1, b_2 \in B$ and $b_1 \vee b_2 \in B$ for any $b_1, b_2 \in B$ such that $b_1 \vee b_2$ exists in L. Note that L^n and B is a modular semilattice if so is L. In the following, let L be a semilattice and B a $(\wedge, \vee)$-closed set in L^n without further mentioning.

For any modular semilattice L, our compact representation theorem is valid for $(\wedge, \vee)$-closed sets in L^n, which are also modular semilattices. However computational problems still remain. As the cardinality of L^n grows exponentially, so may that of $\mathrm{P}(B)$. Moreover, it is unrealistic enumerating $\vee$-irreducible elements of B by a brute-force search. Hirai and Oki [16] solved these problems for $(\wedge, \vee)$-closed sets of S_k. Here S_k is a median semilattice whose underlying set is $\{0\} \cup [k]$ and whose partial order is $0 < i$ for all $i \in [k]$.

We generalize Hirai and Oki's result to $(\wedge, \vee)$-closed sets of arbitrary semilattices. In this section, we give the upper bound of $\mathrm{P}(B)$. The enumerating problem will be treated in the next section. We owe this theorem and its proof to a discussion with Taihei Oki.

Theorem 8 *Let L be a semilattice and B a $(\wedge, \vee)$-closed set in L^n. The cardinality of $\vee$-irreducible elements of B is at most $n|L^{\mathrm{ir}}|$.*

3.2 Constructing PPIP from Menbership Oracle

Let L be a modular semilattice. In this section, we address a polynomial-time algorithm calculating PPIP-representation of B using *Membership Oracle (MO)*.

Definition 9 *Membership Oracle (MO) for a $(\wedge, \vee)$-closed set $B \subseteq L^n$ answers the following decision problem:*

Input: $i, j \in [n]$, $l, l' \in L$,

Output: *Whether or not there exists $\boldsymbol{b} \in B$ such that $\boldsymbol{b}[i] = l$ and $\boldsymbol{b}[j] = l'$.*

An important example of MO is a minimizer oracle. We can show that the minimizer set of a submodular function on L^n forms a $(\wedge, \vee)$-closed set, and that MO of the minimizer set can be reduced to a minimizer oracle. In this sense, it is a natural assumption that MO is available.

Theorem 10 *Suppose that MO is available. Then $\vee$-irreducible elements of B are enumerated by at most $n^2|L|^2$ calls of MO.*

Theorem 11 *The PPIP-representation $P(B)$ can be obtained in $O(n^3|L^{\mathrm{ir}}|^3 + n^2|L|^2)$-time; the algorithm enumerates the partial order, inconsistent relation, and collinear relation of $P(B)$.*

4 Optimal implicational base

In the previous sections, the space complexity to store the relations on a PPIP was ignored. The compact representation by implicational bases deals with this problem. Modular semilattices are sometimes more compactly represented by implicational bases than by PPIPs.

In Section 4.1, we generalize optimal implicational bases for modular lattices, given by Wild [28], for modular semilattices. In Section 4.2, we address a polynomial time algorithm deciding whether a $\cap$-closed system given by implications is a modular semilattice.

4.1 Optimal implicational base for modular semilattice

In this section, we establish a compact representation for a modular semilattice by implicational bases. Our notation given below are a generalization of standard one. See recent survey [29] for more details on closure systems and implications.

We first quickly review $\cap$-closed family and implicational bases. Fix a finite set E. A subset $\mathcal{F} \subseteq 2^E$ is called a *$\cap$-closed family* if $F_1 \cap F_2 \in \mathcal{F}$ for all $F_1, F_2 \in \mathcal{F}$. The members of $\mathcal{F}$ is said to be *closed*.

Modular semilattice L can be viewed as a $\cap$-closed family. In the previous section, we proved that L is isomorphic to a $\cap$-closed family on L^{ir} equipped with inclusion order $\subseteq$, that is, $\mathcal{CS}(\mathrm{P}(L))$ in Theorem 6. A subset $X \subseteq L^{\mathrm{ir}}$ is said to be *inconsistent* if there is no $F \in \mathcal{CS}(\mathrm{P}(L))$ such that $X \subseteq F$.

A pair of subsets $(A, B) \in 2^E \times 2^E$, written as $A \to B$, is called an *implication*. Here A is called the *premise* and B the *conclusion*. An implication is said to be *proper* if its conclusion is nonempty. Let Σ be a collection of implications. We define a $\cap$-closed set $\mathcal{F}(\Sigma) \subseteq 2^E$ as follows: $X \in \mathcal{F}(\Sigma)$ if and only if $A \subseteq X$ implies $B \subseteq X$ for all proper implications $A \to B$ in Σ, and $A \not\subseteq X$ for all improper implications $A \to \emptyset$.

A collection Σ of implications is called an *implicational base* of a $\cap$-closed family $\mathcal{F}$ if $\mathcal{F} = \mathcal{F}(\Sigma)$. The *size* of an implicational base Σ is defined by

$$s(\Sigma) := \sum_{(A \to B) \in \Sigma} (|A| + |B|).$$

An implicational base is said to be *optimal* if its size is minimum among all implicational bases.

Our aim is to give an optimal implicational base for modular semilattice L, viewed as a $\cap$-closed family $\mathcal{CS}(\mathrm{P}(L))$. Let L be a modular semilattice. An element $l' \in L$ is called a *lower cover* of $l \in L$ if $l' < l$ and there is no element $l'' \in L$ such that $l' < l'' < l$. The relation $l' \prec l$ means that l' is a lower cover of l. Every $\vee$-irreducible element q has the unique lower cover $\underline{q}$. If $\underline{q}$ is not the minimum element, then q is said to be *nonatomic*. For every nonatomic element q, its unique lower cover $\underline{q}$ is decomposed by $\vee$-irreducible elements $\{p_i\}$ as $\underline{q} = p_1 \vee p_2 \vee \cdots \vee p_n$. The sequence $\{p_i\}$ is called an *irreducible decomposition* of $\underline{q}$ if no proper subsequence $\{p_{i_k}\}$ decomposes $\underline{q}$, i.e., satisfies $\underline{q} = p_{i_1} \vee p_{i_2} \vee \cdots \vee p_{i_m}$. For a nonatomic $\vee$-irreducible element q, let $B_q = \{p_1, p_2, \ldots, p_m\}$ denote an irreducible decomposition of $\underline{q}$. An element $l \in L$ is called an *$\mathcal{M}_n$-element* ($n \geq 3$) if there are $y \prec x_0, x_1, \ldots, x_{n-1} \prec l$ in L such that $x_i \wedge x_j = y$ and $x_i \vee x_j = l$ for all distinct i, j in $\{0, 1, \ldots, n-1\}$. We call y a bottom and x_i an intermediate element. A function $\phi\colon L \to 2^{L^{\mathrm{ir}}}$ is defined by $\phi(l) = \{p \in L^{\mathrm{ir}} \mid p \leq l\}$.

Wild [28] characterized an optimal implicational base for a modular lattice.

Theorem 12 ([28], PROPOSITION 5) *Let L be a modular lattice. An optimal implicational base for $\mathcal{CS}(\mathrm{P}(L))$ consists of the following implications:*

- *$\{q\} \to B^q$ for every nonatomic $q \in L^{\mathrm{ir}}$;*
- *$\{p_i^x, q_j^x\} \to \{r^x_{j+1 \mod n}\}$ for all $0 \leq i < j \leq n-1$ and $\mathcal{M}_n$-elements $x \in L$ with the bottom y and intermediate elements $x_0, x_1, \ldots, x_{n-1}$, where $p_i^x \in \phi(x_i) \setminus \phi(y)$, $q_j^x \in \phi(x_j) \setminus \phi(y)$, and $r^x_{j+1 \mod n} \in \phi(x_{j+1 \mod n}) \setminus \phi(y)$;*

We generalize his result for a modular semilattice. A pair $(p, q) \in L^{\mathrm{ir}} \times L^{\mathrm{ir}}$ is called a *minimal inconsistent pair* if the following conditions are satisfied: $p \vee q$ dose not exist; if $p' \leq p$, $q' \leq q$, and $p' \vee q'$ does not exist, then $p = p'$ and $q = q'$ for any $p', q' \in L^{\mathrm{ir}}$.

Theorem 13 *Let L be a modular semilattice. An optimal implicational base for $\mathcal{CS}(\mathrm{P}(L))$ consists of the following implications:*

- $\{q\} \to B^q$ *for every nonatomic* $q \in L^{\mathrm{ir}}$*;*
- $\{p_i^x, q_j^x\} \to \{r_{j+1 \mod n}^x\}$ *for all* $0 \leq i < j \leq n-1$ *and* $\mathcal{M}_n$*-elements* $x \in L$ *with the bottom* y *and intermediate elements* $x_0, x_1, \ldots, x_{n-1}$*, where* $p_i^x \in \phi(x_i) \setminus \phi(y)$*,* $q_j^x \in \phi(x_j) \setminus \phi(y)$*, and* $r_{j+1 \mod n}^x \in \phi(x_{j+1 \mod n}) \setminus \phi(y)$*;*
- $\{p, q\} \to \emptyset$ *for every minimal inconsistent pair* $(p, q) \in L^{\mathrm{ir}} \times L^{\mathrm{ir}}$*.*

Compact representation by implicational bases is efficient when the modular semilattice L contains large *diamond*. *Diamond* is a modular lattice whose height is two and whose maximum element is an $\mathcal{M}_n$-element. To represent a diamond by a PPIP, we need $O(n^3)$ collinear triples. However optimal implicational base for it contains $O(n^2)$ implications.

Example 14 *An optimal implicational base for the modular semilattice in Figure 1 (a) consists of the following implications:*

$$
\begin{aligned}
4 &\to 5,\\
5 &\to 1, 3,\\
6 &\to 1, 2,\\
7 &\to 2, 3,\\
5, 6 &\to 7,\\
6, 7 &\to 5,\\
7, 5 &\to 6,\\
1, 8 &\to \emptyset,\\
2, 4 &\to \emptyset.
\end{aligned}
$$

4.2 Identifying modular semilattice

We generalize Herrmann and Wild's algorithm [12] deciding whether a closure system given by implications is a modular lattice.

Theorem 15 *Let Σ be the family of implications on E. We can decide whether or not $\mathcal{F}(\Sigma)$ is modular in $O((s(\Sigma)|\Sigma|^2|E|^4 + |E|^7) \log |E|)$-time.*

Acknowledgment

We thank Taihei Oki for helpful comment and discussion, especially for Theorem 8. This research is supported by JSPS KAKENHI Grant Numbers, 25280004, 26330023, 26280004.

References

[1] M. Aigner, *Combinatorial Theory*, Springer-Verlag, Berlin (1997). Reprint of the 1979 original.

[2] F. Ardila, M. Owen and S. Sullivant, Geodesics in CAT(0) cubical complexes, *Advances in Applied Mathematics*, **48**(1):142–163 (2012).

[3] M. Arias and J. L. Balcázar, Canonical Horn representations and query learning, In *Algorithmic Learning Theory* (20th International Conference on Algorithmic Learning Theory, ALT 2009) [Lecture Notes in Computer Science 5809], pp. 156–170, Springer, Berlin, (2009).

[4] H.-J. Bandelt, M. Van de Vel, and E. Verheul, Modular interval space, *Mathematische Nachrichten* **163**:177–201 (1993).

[5] J.-P. Barthélemy and J. Constantin, Median graphs, parallelism and posets, *Discrete Mathematics* **111**(1-3):49–63 (1993).

[6] J. Berman, P. Idziak, P. Marković, R. McKenzie, M. Valeriote, and R. Willard, Varieties with few subalgebras of powers, *Transactions of the American Mathematical Society.* , **362**(3):1445–1473 (2010).

[7] N. Buchbinder, M. Feldman, J. Seffi, R. Schwartz, A tight linear time (1/2)-approximation for unconstrained submodular maximization, *SIAM Journal on Computing*, **44**(5):1384–1402 (2015).

[8] S. Burris and H. P. Sankappanavar, *A Course in Universal Algebra*, Springer-Verlag, New York-Berlin (1981).

[9] S. Fujishige, *Submodular Functions and Optimization*, 2nd ed, Elsevier B. V., Amsterdam (2005).

[10] I. Gridchyn and V. Kolmogorov, Potts model, parametric maxow and k-submodularfunctions, In *Proceedings of the IEEE International Conference on Computer Vision(ICCV 2013)*, pp 2320–2327 (2013).

[11] C. Herrmann, D. Pickering, and M. Roddy, A geometric description of modular lattices, *Algebra Universalis*, **31**(3):365–396 (1994).

[12] C. Herrmann and M. Wild, A polynomial algorithm for testing congruence modularity, *International Journal of Algebra and Computation*, **6**(4):379–387 (1996).

[13] H. Hirai, Computing DM-decomposition of a partitioned matrix with rank-1 blocks, `arXiv:1609.01934` (2016).

[14] H. Hirai, Discrete convexity and polynomial solvability in minimum 0-extension problems, *Mathematical Programming, Series A*, **155**:1-55 (2016).

[15] H. Hirai, L-convexity on graph structures, `arXiv:1610.02469` (2016).

[16] H. Hirai and T. Oki, A compact representation for minimizers of k-submodular functions, In *Combinatorial Optimization* (4th International Symposium on Combinatorial Optimization, ISCO 2016) [Lecture Notes in Computer Science 9849], pp. 381–392, Springer International Publishing, Switzerland (2016).

[17] H. Hirai and Y. Iwamasa, On k-submodular relaxation, *SIAM Journal on Discrete Mathematics*, **30**(3):1726–1736 (2016).

[18] A. Huber and V. Kolmogorov, Towards minimizing k-submodular functions, In *Proceedings of the 2nd International Symposium on Combinatorial Optimization (ISCO2012)*, [Lecture Notes in Computer Science 7422], pp. 451-462, Springer, Heidelberg (2012).

[19] H. Ito, S. Iwata and K. Murota, Block-triangularizations of partitioned matrices under similarity/equivalence transformations, *SIAM Journal on Matrix Analysis and Applications*, **15**(4):1226–1255 (1994).

[20] V. Kolmogorov, J. Thapper and S. Živný, The power of linear programming for general-valued CSPs, *SIAM Journal on Computing*, **44**(1):1–36 (2015).

[21] D. Maier, *The Theory of Relational Databases*, Computer Science Press, USA (1983).

[22] K. Murota, *Matrices and Matroids for Systems Analysis*, Springer-Verlag, Berlin (2000).

[23] K. Murota, M. Iri and M. Nakamura, Combinatorial canonical form of layered mixed matrices and its application to block-triangularization of systems of linear/nonlinear equations, *SIAM Journal on Algebraic Discrete Methods*, **8**(1):123–149 (1987).

[24] G. L. Nemhauser, L. A. Wolsey, M. L. Fisher, An analysis of approximations for maximizing submodular set functions. I *Mathematical Programming*, **14**(3):265–294 (1978).

[25] M. Nielsen, G. Plotkin, G. Winskel, Petri nets, event structures and domains, part I, *Theoretical Computer Science*, **13**:85-108 (1981).

[26] M. Sholander, Medians and betweenness, In *proceedings of the American Mathematical Society*, **5**(5):801-807 (1954).

[27] J. Ueberberg, *Foundations of Incidence Geometry*, Springer, Heidelberg (2011).

[28] M. Wild, Optimal implicational bases for finite modular lattices, *Quaestiones Mathematicae* **23**(2):153–161 (2000).

[29] M. Wild, The joy of implications, aka pure Horn formulas: Mainly a survey, *Theoretical Computer Science*, **658**(part B):264–292 (2017).

Reconfiguring Optimal Ladder Lotteries

Takashi Horiyama

Saitama University, Saitama, Japan
horiyama@al.ics.saitama-u.ac.jp

Kunihiro Wasa

National Institute of Informatics,
Chiyoda, Japan
wasa@nii.ac.jp

Katsuhisa Yamanaka

Iwate University, Morioka, Japan
yamanaka@cis.iwate-u.ac.jp

Abstract: A ladder lottery, known as "Amidakuji" in Japan, is a common way to decide an assignment at random. A ladder lottery L of a given permutation is optimal if L has the minimum number of horizontal lines. In this paper, we investigate a reconfiguration problem of optimal ladder lotteries. The reconfiguration problem on a set of optimal ladder lotteries asks, given two optimal ladder lotteries L, L' of a permutation π, to find a sequence of $\langle L_1, L_2, \ldots, L_k \rangle$ of optimal ladder lotteries of π such that (1) $L_1 = L$ and $L_k = L'$ and (2) L_i for $i = 2, 3, \ldots, m$ is obtained from L_{i-1} by moving a bar in L_{i-1} locally. An existing result implies that any two optimal ladder lotteries of a permutation π have a reconfiguration sequence of length $O(n^3)$, where n is the number of elements in π. In this paper, we propose an exact formula for the minimum length of reconfiguration sequences between two optimal ladder lotteries.

Keywords: reconfiguration problem, ladder lottery, optimal ladder lottery

1 Introduction

A *ladder lottery*, known as the "Amidakuji" in Japan, is a common way to decide an assignment at random. Formally, a ladder lottery L of a permutation $\pi = (p_1, p_2, \ldots, p_n)$ is a network with n vertical lines (*lines* for short) and zero or more horizontal lines (*bars* for short) each of which connects two consecutive vertical lines. The i-th line from the left is called *line* i. The top ends of lines correspond to π. The bottom ends of lines correspond to the identical permutation $(1, 2, \ldots, n)$. See Figure 1. Each element p_i in π starts the top end of line i, and goes down along the line, then whenever p_i comes to an endpoint of a bar, p_i goes horizontally along the bar to the other end, then goes down again. Finally p_i reaches the bottom end of line p_i. We can regard a bar as a modification of the current permutation, and a

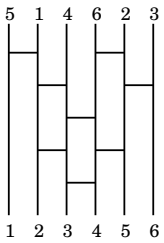


Figure 1: An optimal ladder lottery of the permutation (5,1,4,6,2,3).

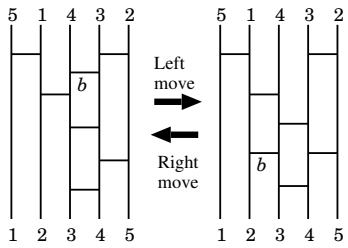


Figure 2: Movements of bars. A left move of b in the left ladder lottery is applied. The right move of b in the right ladder lottery is applied.

sequence of such modifications in a ladder lottery always results in the identical permutation $(1, 2, \ldots, n)$. A ladder lottery of a permutation $\pi = (p_1, p_2, \ldots, p_n)$ is *optimal* if it consists of the minimum number of bars among ladder lotteries of π. Let L be an optimal ladder lottery of π and m be the number of bars in L. Then we can observe that m is equal to the number of "inversions" of π, which is a pair (p_i, p_j) in π with $p_i > p_j$ and $i < j$. The ladder lottery in Figure 1 has 8 bars and the permutation (5,1,4,6,2,3) has 8 inversions: (5,1), (5,4), (5,2), (5,3), (4,2), (4,3), (6,2), and (6,3), so the ladder lottery is optimal. The ladder lotteries are related to some objects in theoretical computer science. First, the ladder lotteries are strongly related to primitive sorting networks, which are deeply investigated by Knuth [3]. A comparator in a primitive sorting network replaces p_i and p_{i+1} by $\min(p_i, p_{i+1})$ and $\max(p_i, p_{i+1})$, while a bar in a ladder lottery always exchanges them. Next, the set of the optimal ladder lotteries of a reverse permutation one-to-one corresponds to arrangements of pseudolines. Each line in a ladder lottery corresponds to a pseudoline in an arrangement, and each bar corresponds to an intersection of two pseudolines.

In this paper, we investigate a reconfiguration problem of optimal ladder lotteries. A reconfiguration problem on a set S asks, given two elements $e, e' \in S$, whether or not there exists a sequence $\langle e_1, e_2, \ldots, e_k \rangle$ of elements in S such that (1) $e_1 = e$ and $e_k = e'$ and (2) e_i for $i = 2, 3, \ldots, k$ is obtained from e_{i-1} with a designated unit operation. Recently, reconfiguration problems have been extensively studied [1, 2, 6, 7]. A reconfiguration problem of optimal ladder lotteries asks, given two optimal ladder lotteries L and L', to calculate the minimum number of "bar movements" required to obtain L' from L. A *left move* of a bar is an operation for the bar in a ladder lottery. Intuitively, the left move of a bar b is to move b to the lower-left position so that the corresponding permutation does not change. Similarly, a *right move* of a bar b is to move b to the upper-right position so that the corresponding permutation does not change. See Figure 2. The formal definitions of the two operations are given in the next section. Note that, each of the bar movements in a ladder lottery generates a different ladder lottery, but the two ladder lotteries correspond to the same permutation (the moved bar is required to satisfy some conditions).

It is known that, for two optimal ladder lotteries L and L' of a permutation, L can be obtained from L' by repeatedly applying left or right moves of bars [5]. Yamanaka *et al.* [8] implicitly showed that the length of such sequences of bar movements is bounded by at most n^3. They defined a rooted tree structure on the set of optimal ladder lotteries of a permutation π such that (1) each node corresponds to an optimal ladder lottery of π and (2) each edge corresponds to either a left move or right move. The depth of the tree is at most $\frac{n^3}{2}$, where n is the number of elements in π, thus we always have a sequence of length at most n^3 between any two optimal ladder lotteries. However, the sequences have redundant bar moves for some instances. For example, for the two optimal ladder lotteries L and L' in Figure 3, the reconfiguration sequence by Yamanaka *et al.* between L and L' is shown in the figure. The length of the sequence is 9. However, the minimum length is only 1, since L' is obtained by only applying a right move to the bar b in L.

In this paper, we investigate the minimum length of reconfiguration sequences of optimal ladder lotteries. To best of our knowledge, there is no result to calculate the minimum length of reconfiguration

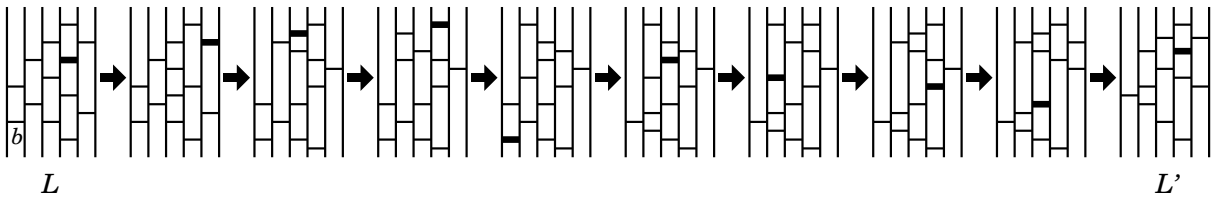


Figure 3: A sequence of optimal ladder lotteries between L and L' of $(5, 6, 3, 4, 2, 1)$. Each ladder lottery except L is obtained by locally moving the thick bar in the left adjacent ladder lottery.

sequences between two optimal ladder lotteries. We propose an exact formula for the minimum length and a polynomial-time algorithm that finds a minimum-length reconfiguration sequence.

2 Preliminary

A *ladder lottery* L of a permutation $\pi = (p_1, p_2, \ldots, p_n)$ is a network with n vertical lines (*lines* for short) and zero or more horizontal lines (*bars* for short) each of which connects two consecutive vertical lines. The i-th line from the left is called *line* i. The top ends of the n lines correspond to π. The bottom ends of the n lines correspond to the identical permutation $(1, 2, \ldots, n)$. See Figure 1. Each element p_i in π starts the top end of line i, and goes down along the line, then whenever p_i comes to an endpoint of a bar p_i goes to the other end and goes down again, then finally p_i reaches the bottom end of line p_i. This path is called the *route* of the element p_i. The route of p_i divides L into the two regions. For the route of p_i, the *left region* is the left side of the route, which includes the top ends of $p_1, p_2, \ldots, p_{i-1}$ and the bottom ends of $1, 2, \ldots, p_i - 1$. Similarly, the *right region* is the right side of the route, which includes the top ends of $p_{i+1}, p_{i+2}, \ldots, p_n$ and the bottom ends of $p_i + 1, p_i + 2, \ldots, n$. The left region and right region of the route are the properly left side and right side of the route, respectively. We can regard a bar as a modification of the current permutation, and a sequence of such modifications in a ladder lottery always results in the identical permutation $(1, 2, \ldots, n)$. Let $\pi = (p_1, p_2, \ldots, p_n)$ be a permutation. An *inversion* of π is a pair (p_i, p_j) with $p_i > p_j$ and $i < j$. Let m be the number of inversions of π. We can observe that any ladder lottery of π contains at least m bars, since each bar "cancels" at most one inversion of the "current" permutation (see, e.g., [4, 5.3.4 Figure 45]). If a ladder lottery L contains exactly m bars, then we say that L is *optimal*.

We denote by $[1, n]$ the set $\{1, 2, \ldots, n\}$. Let π be a permutation of $[1, n]$. We denote by $\mathcal{L}(\pi)$ the set of optimal ladder lotteries of π. Let L be an optimal ladder lottery in $\mathcal{L}(\pi)$. For an element x in π, we denote by $R_\ell(L, x)$ the left region of the route of x in L. Similarly, we denote by $R_r(L, x)$ the right region of the route of x in L. Two endpoints of two distinct bars are *visible* if the line segment between the two endpoints includes no other endpoint. For an element z in π, we denote by $BS(z) = \langle b_1, b_2, \ldots, b_p \rangle$ the sequence of bars on the route of z from top to bottom. A pair (b_i, b_{i+1}) of two consecutive bars in $BS(z)$ is *lower-right* if the right endpoint of b_i and the left endpoint of b_{i+1} are on the same line. Note that the two endpoints are visible from the definition of the routes. Similarly, (b_i, b_{i+1}) is *lower-left* if the left endpoint of b_i and the right endpoint of b_{i+1} are on the same line.

In an optimal ladder lottery L, any pair of two elements is swapped at most once in a bar from optimality. Thus, a bar in L can be uniquely labeled as a pair of the two elements which are swapped in the bar. Let $\{x, y\}$ be a bar in $R_\ell(L, z)$. Then, a triple $(x, y; z)$, $x, y, z \in [1, n]$, is *right-movable* if either of the following two conditions holds:

(1) For some lower-right pair (b_i, b_{i+1}) in $BS(z)$, the left endpoint of $\{x, y\}$ is visible from the left endpoint of b_i and the right endpoint of $\{x, y\}$ is visible from the left endpoint of b_{i+1}. See Figure 4(a).

(2) For some lower-left pair (b_i, b_{i+1}) in $BS(z)$, the left endpoint of $\{x, y\}$ is visible from the left endpoint of b_{i+1} and the right endpoint of $\{x, y\}$ is visible from the left endpoint of b_i. See

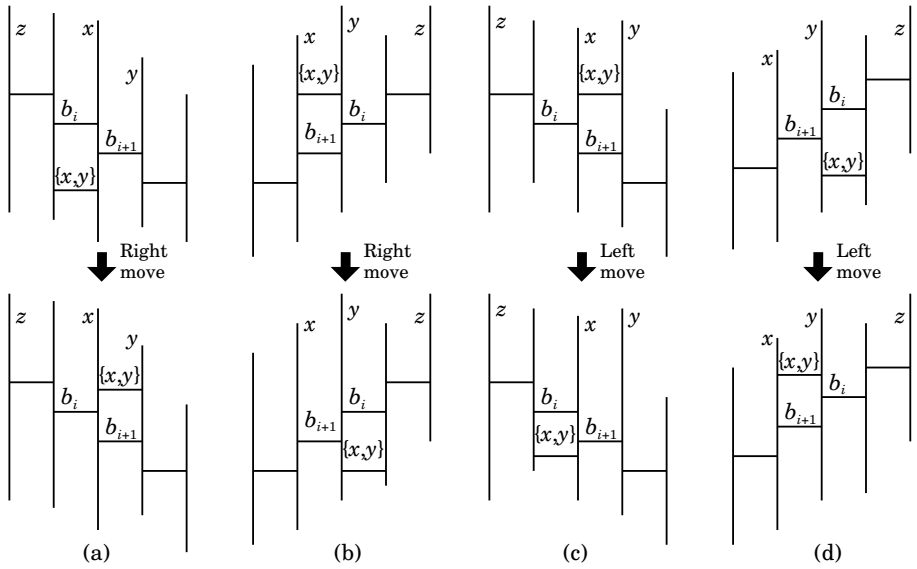


Figure 4: The triple $(x, y; z)$ is right-movable in (a) and (b) and is left-movable in (c) and (d).

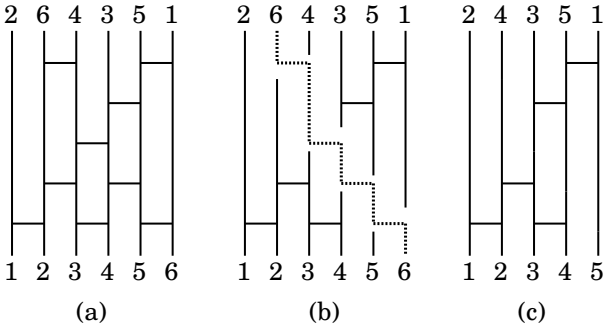


Figure 5: The removal of the route of n. (a) The original optimal ladder lottery L. (b) The removal of the route of the element $n = 6$. (c) Patching $R_\ell(L, n)$ with $R_r(L, n)$.

Figure 4(b).
When the triple $(x, y; z)$ satisfies (1), the *right move* is to move $\{x, y\}$ to the upper-right beyond the route of z such that the left endpoint of $\{x, y\}$ is visible from the right endpoint of b_i and the right endpoint of $\{x, y\}$ is visible from the right endpoint of b_{i+1}(Figure 4(a)). When the triple $(x, y; z)$ satisfies (2), the *right move* is to move $\{x, y\}$ to the lower-right beyond the route of z such that the left endpoint of $\{x, y\}$ is visible from the right endpoint of b_{i+1} and the right endpoint of $\{x, y\}$ is visible from the right endpoint of b_i(Figure 4(b)). Similarly, for the case that $\{x, y\}$ is in $R_r(L, z)$, we define a *left-movability* and a *left move* of a triple $(x, y; z)$. See Figure 4(c) and (d). Note that the ladder lottery obtained from L by either a left move or right move corresponds to the same permutation as L.

Now, let us consider to remove the route of n from L, and then obtain an optimal ladder lottery with one less line. Recall that the route of n partitions L into $R_\ell(L, n)$ and $R_r(L, n)$. Removing the route of n from L then patching $R_\ell(L, n)$ with $R_r(L, n)$, as shown in Figure 5, results in an optimal ladder lottery with $n - 1$ lines of the permutation obtained from π by removing n.

Let L, L' be two ladder lotteries in $\mathcal{L}(\pi)$. A sequence $\langle L_1, L_2, \ldots, L_k \rangle$ of ladder lotteries in $\mathcal{L}(\pi)$ is a *reconfiguration sequence* between L and L' if the following two conditions hold:
(1) $L_1 = L$ and $L_k = L'$,

(2) L_i is obtained from L_{i-1} by applying either a left move or right move to a bar in L_{i-1} for $i = 2, 3, \ldots, k$.

The *length* of a reconfiguration sequence is the number of the ladder lotteries in the sequence minus one. That is, the length of a reconfiguration sequence is the number of bar movements to obtain L' from L. We denote by $\mathrm{OPT}(L, L')$ the minimum length of reconfiguration sequences between L and L' in $\mathcal{L}(\pi)$. A *reconfiguration problem of optimal ladder lotteries* asks to calculate $\mathrm{OPT}(L, L')$ for given two ladder lotteries L, L' in $\mathcal{L}(\pi)$. Yamanaka *et al.* [8] implicitly showed the following upper bound of $\mathrm{OPT}(L, L')$.

Theorem 1 ([8]) *Let π be a permutation in $[1, n]$. For two ladder lotteries L, L' in $\mathcal{L}(\pi)$, $OPT(L, L') \leq n^3$ holds.*

3 Lower bound

Let π be a permutation of $[1, n]$. Let L, L' be two ladder lotteries in $\mathcal{L}(\pi)$. In this section, we give a lower bound of $\mathrm{OPT}(L, L')$.

Let $\{x, y\}$ be a bar in L. A triple $(x, y; z)$, $x, y, z \in \pi$, is *reverse* if (1) L includes the bar $\{x, y\}$ and (2) either $\{x, y\}$ is in $R_r(L, z)$ and is in $R_\ell(L', z)$ or $\{x, y\}$ is in $R_\ell(L, z)$ and is in $R_r(L', z)$. Intuitively, for a reverse triple $(x, y; z)$, $\{x, y\}$ must be moved beyond the route of z to obtain L' from L. We denote by $\#rev(L, L')$ the number of the reverse triples for L and L'. Note that $L = L'$ holds if and only if $\#rev(L, L') = 0$ holds. From the definition, each of a left move and right move decreases or increases $\#rev(L, L')$ by one. Hence, one can observe the following lemma.

Lemma 2 *Let π be a permutation of $[1, n]$, and let L, L' be two ladder lotteries in $\mathcal{L}(\pi)$. Then, $OPT(L, L') \geq \#rev(L, L')$ holds.*

4 Upper bound

Let π be a permutation of $[1, n]$, and let L, L' be two ladder lotteries in $\mathcal{L}(\pi)$. In this section we give an upper bound of $\mathrm{OPT}(L, L')$:

Lemma 3 *Let π be a permutation of $[1, n]$, and let L, L' be two ladder lotteries in $\mathcal{L}(\pi)$. Then, $OPT(L, L') \leq \#rev(L, L')$ holds.*

To show the lemma, we prove that there always exists a triple $(x, y; z)$, $x, y, z \in \pi$, such that either a left move or right move of $\{x, y\}$ decreases $\#rev(L, L')$ in the rest of this section.

A triple $(x, y; z)$ is *movable* in L if $(x, y; z)$ is either right-movable or left-movable. Suppose $(x, y; z)$ is movable in L. Let M be the ladder lottery obtained from L by applying either a left move or right move to $\{x, y\}$. The movable triple $(x, y; z)$ is *improving* if $\#rev(M, L') = \#rev(L, L') - 1$ holds. Intuitively, for an improving triple $(x, y; z)$, applying a bar-movement to $\{x, y\}$ decreases $\#rev(L, L')$. The following lemma immediately implies Lemma 3.

Lemma 4 *Let π be a permutation of $[1, n]$, and let L, L' be two distinct ladder lotteries in $\mathcal{L}(\pi)$. There exists an improving triple in L.*

Proof: We prove the claim by induction on the number of lines, namely the number of elements in a permutation. If the number of lines in an optimal ladder lottery is 3, then the claim holds clearly. Now we assume that the claim holds for optimal ladder lotteries with $n - 1$ lines.

Let π' be the permutation obtained from π by removing n. Let K and K' be ladder lotteries obtained from L and L' by removing the route of n, respectively. If K and K' are equivalent, then we can find an improving triple, as in the following claim.

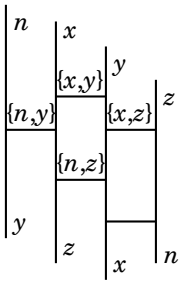


Figure 6: A local configuration in L.

Claim 5 *Let L and L' be two distinct optimal ladder lotteries of a permutation of $[1, n]$, let K and K' be ladder lotteries obtained from L and L' by removing the route of n, respectively. Suppose that K and K' are equivalent. Then there exists an improving triple in L.*

Proof: If K and K' are identical but L and L' are not identical, without loss of generality, there exists a bar $\{x, y\}$ in $R_r(L, n)$ and in $R_\ell(L', n)$ such that the left endpoint of $\{x, y\}$ in L is visible from the right endpoint of a bar $\{n, y\}$ in $BS(n)$. Let $\{n, z\}$ be the next bar of $\{n, y\}$ in $BS(n)$. Now, we consider the following two cases.

(1) If the right endpoints of $\{x, y\}$ and $\{n, z\}$ are mutually visible, then $(x, y; n)$ is improving. (2) Suppose that there exists a bar $\{x, z\}$ in L such that its left endpoint is visible from the right endpoints of $\{x, y\}$ and $\{n, z\}$ (see Figure 6).

Without loss of the generality, the right endpoint of $\{x, z\}$ is visible from the right endpoint of a bar in $BS(n)$. (If we assume otherwise, we can find such a bar in L.) (2.1) If $\{x, z\}$ is in $R_\ell(L', n)$, then $(x, z; n)$ is improving. (2.2) Suppose that $\{x, z\}$ is in $R_r(L', n)$. (2.2.1) If $\{x, y\}$ precedes $\{x, z\}$ in $BS(x)$ in L', the route of x crosses the route of n more than once. (2.2.2) If $\{x, y\}$ succeeds $\{x, z\}$ in $BS(x)$ in L', the route of x crosses the route of z more than once. When a route crosses another route more than once, there exist more than one bar whose endpoints are the same. Thus, (2.2.1) and (2.2.2) imply that L' is not optimal, and (2.2) contradicts the assumption. Therefore, there always exists an improving triple in L. □

Thus, we can assume K and K' are distinct. Since K, K' in $\mathcal{L}(\pi')$, from the induction hypothesis, there exists an improving triple $(x, y; z)$. If $(x, y; z)$ is also improving in L, the claim holds. Thus, we assume that $(x, y; z)$ is not improving in L. If $(x, y; z)$ is movable but not improving in L, this contradicts to the assumption that $(x, y; z)$ is improving in K. Therefore, we assume that $(x, y; z)$ is neither improving nor movable in L.

We have the following case analysis. First, we analyze the case that $(x, y; z)$ is left-movable in K. Then the local configuration around $(x, y; z)$ in K is illustrated in Figure 7(a).

Case 1: $(x, y; z)$ is left-movable in K.

Let us consider all the possible patterns of L and L' in this case. To discuss all the patterns, we give notations as illustrated in Figure 7(a). We define the 5 regions in K: $R_1 = R_\ell(K, x) \cap R_\ell(K, z)$, $R_2 = R_r(K, x) \cap R_\ell(K, y) \cap R_\ell(K, z)$, $R_3 = R_r(K, y) \cap R_\ell(K, z)$, $R_4 = R_r(K, x) \cap R_r(K, y) \cap R_r(K, z)$, and $R_5 = R_r(K, x) \cap R_\ell(K, y) \cap R_r(K, z)$. We also define the 3 line segments included in the route of x: (1) x_1 is the line segment such that its bottom endpoint is the left endpoint of $\{x, z\}$, (2) x_2 is the line segment such that its top endpoint is the right endpoint of $\{x, z\}$ and its bottom endpoint is the left endpoint of $\{x, y\}$, and (3) x_3 is the line segment such that its top endpoint is the right endpoint of $\{x, y\}$. Similarly, as shown in Figure 7, we define y_1, y_2, y_3, z_1, z_2, and z_3.

Now, let us consider all possible patterns of L. We have a case analysis on the route of n in L. If the route of n does not pass through R_5 in K, then $(x, y; z)$ is left-movable and hence improving in L. Hence, in what follows, we consider the other cases.

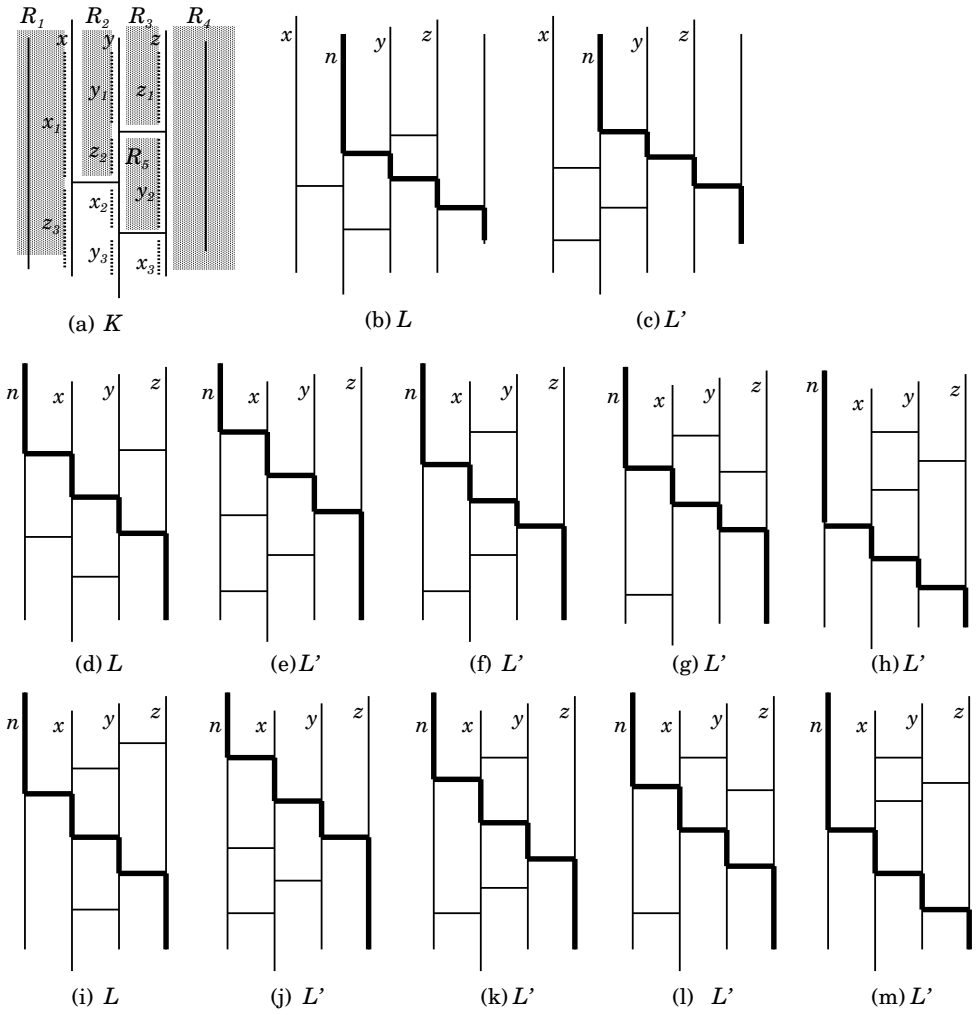


Figure 7: Illustration for Case 1. In L', all the routes except x, y, z, and n are removed for simplicity.

Case 1-1: The route of n passes through R_2 and then it has intersections with z_2 and y_2 in this order (See Figure 7(b)).

In this case, all the possible pattern of the routes of x, y, z, and n in L' is only one pattern, as shown in Figure 7(c). In the figure, all the routes except x, y, z, and n are omitted for simplicity. (Note that configurations consisting only the 4 routes in L' are sufficient for this analysis.) Then, $(y, z; n)$ is improving in L.

Case 1-2: The route of n passes through R_1 and then it has intersections with x_1, z_2, and y_2 in this order (See Figure 7(d)).

All the possible patterns of the routes of x, y, z, and n in L' are the 4 patterns, as shown in Figure 7(e), (f), (g), and (h). Then, $(y, z; n)$ is improving in L for Figure 7(e), (f), and (g). $(x, z; n)$ is improving in L for Figure 7(g) and (h).

Case 1-3: The route of n passes through R_1 and then it has intersections with z_3, x_2, and y_2 in this order (See Figure 7(i)).

All possible patterns of the routes of x, y, z, and n in L' are 4 patterns, as shown in Figure 7(j), (k), (l), and (m). Then, $(x, z; n)$ is improving in L for Figure 7(j) and (k). $(x, y; n)$ is improving in L for

Figure 7(k), (l), and (m).

Next, we analyze the case that $(x, y; z)$ is right-movable.

Case 2: $(x, y; z)$ is right-movable in K.
We have a similar case analysis as Case 1. We omit the detail in this manuscript. □

From Lemma 2 and Lemma 3, we have the following theorem.

Theorem 6 *Let π be a permutation of $[1, n]$, and let L, L' be two ladder lotteries in $\mathcal{L}(\pi)$. Then, $OPT(L, L') = \#rev(L, L')$ holds.*

From Lemma 4, one can find an improving triple for any two ladder lotteries L, L'. This can be done in polynomial time. Besides, $\mathrm{OPT}(L, L')$ is polynomial from Theorem 6. Hence, we have the following corollary.

Corollary 7 *Let π be a permutation of $[1, n]$, and let L, L' be two ladder lotteries in $\mathcal{L}(\pi)$. One can find a reconfiguration sequence of the minimum length between L and L' in polynomial time.*

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers JP24106007, JP15H05711, JP15K00008, JP15KT0020, JP16K00002.

References

[1] Parikshit Gopalan, Phokion G. Kolaitis, Elitza Maneva, and Christos H. Papadimitriou. The connectivity of boolean satisfiability: Computational and structural dichotomies. *SIAM Journal on Computing*, 38(6):2330–2355, 2009.

[2] Takehiro Ito, Erik D. Demaine, Nicholas J.A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12):1054 – 1065, 2011.

[3] Donald E. Knuth. *Axioms and hulls.* LNCS 606, Springer-Verlag, 1992.

[4] Donald E. Knuth. *The art of computer programming*, volume 3. Addison-Wesley, 2nd edition, 1998.

[5] Laurent Manivel. *Symmetric Functions, Schubert Polynomials and Degeneracy Loci.* American Mathematical Soc., 2001.

[6] Amer E. Mouawad, Naomi Nishimura, Vinayak Pathak, and Venkatesh Raman. Shortest reconfiguration paths in the solution space of boolean formulas. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming*, volume 9134 of *Lecture Notes in Computer Science*, page 985996, 2015.

[7] Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, pages 1–24, 2016.

[8] K. Yamanaka, S. Nakano, Y. Matsui, R. Uehara, and K. Nakada. Efficient enumeration of all ladder lotteries and its application. *Theoretical Computer Science*, 411:1714–1722, 2010.

Streaming Submodular Maximization under a Knapsack Constraint

Chien-Chung Huang

Département d'informatique,
École Normale Supérieure,
45, rue d'Ulm, Paris, France.
villars@gmail.com

Naonori Kakimura[1]

Department of Mathematics,
Keio University
Yokohama 223-8522, Japan.
kakimura@math.keio.ac.jp

Yuichi Yoshida[2]

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo
101-8430, Japan
yyoshida@nii.ac.jp

Abstract: In this paper, we consider the problem of maximizing a monotone submodular function subject to a knapsack constraint in the streaming setting. In particular, the elements arrive sequentially and at any point of time, the algorithm has access only to a small fraction of the data stored in primary memory. For this problem, we propose a $(0.363-\varepsilon)$-approximation algorithm, requiring only a single pass through the data; moreover, we propose a $(0.4-\varepsilon)$-approximation algorithm requiring a constant number of passes through the data. The required memory space of both algorithms depends only on the size of the knapsack capacity and ε.

Keywords: Submodular function, Single-pass streaming algorithm, Constant approximation

1 Introduction

A set function $f : 2^E \to \mathbb{R}_+$ on a ground set E is called *submodular* if it satisfies the *diminishing marginal return property*, i.e., for any subsets $S \subseteq T \subsetneq E$ and $e \in E \setminus T$, we have

$$f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(T).$$

A function is *monotone* if $f(S) \leq f(T)$ for any $S \subseteq T$. Submodular functions play a fundamental role in combinatorial optimization, as they capture rank functions of matroids, edge cuts of graphs, and set coverage, just to name a few examples. Besides their theoretical interests, submodular functions have attracted much attention from the machine learning community because they can model various practical problems such as online advertising [1, 11, 18], sensor location [12], text summarization [17, 16], and maximum entropy sampling [14].

Many of the aforementioned applications can be formulated as the maximization of a monotone submodular function under a knapsack constraint. In this problem, we are given a monotone submodular

[1]Partly supported by JST ERATO Grant Number JPMJER1305, and JSPS KAKENHI Grant Numbers JP25730001, JP24106002, and JP17K00028.
[2]Supported by JST ERATO Grant Number JPMJER1305.

function $f : 2^E \to \mathbb{R}_+$, a size function $c : E \to \mathbb{N}$, and an integer $K \in \mathbb{N}$, where $\mathbb{N}$ denotes the set of positive integers. The problem is defined as

$$\text{maximize} \quad f(S) \quad \text{subject to} \quad c(S) \leq K, \tag{1}$$

where we denote $c(S) = \sum_{e \in S} c(e)$ for a subset $S \subseteq E$. Throughout this paper, we assume that every item $e \in E$ satisfies $c(e) \leq K$ as otherwise we can simply discard it. Note that, when $c(e) = 1$ for every item $e \in E$, the constraint coincides with a cardinality constraint.

The problem of maximizing a monotone submodular function under a knapsack constraint is classical and well-studied. First introduced by Wolsey [20], the problem is known to be NP-hard but can be approximated within the factor of (close to) $1 - 1/e$; see e.g., [3, 10, 13, 8, 19].

In some applications, the amount of input data is much larger than the main memory capacity of individual computers. In such a case, we need to process data in a *streaming* fashion. That is, we consider the situation where each item in the ground set E arrives sequentially, and we are allowed to keep only a small number of the items in memory at any point. This setting effectively rules out most of the techniques in the literature, as they typically require random access to the data. In this work, we also assume that the function oracle of f is available at any point of the process. Such an assumption is standard in the submodular function literature and in the context of streaming setting [2, 7, 21]. Badanidiyuru *et al.* [2] discuss several interesting and useful functions where the oracle can be implemented using a small subset of the entire ground set E.

We note that the problem, under the streaming model, has so far not received its deserved attention in the community. Prior to the present work, we are aware of only two: for the special case of cardinality constraint, Badanidiyuru *et al.* [2] gave a single-pass $(1/2 - \varepsilon)$-approximation algorithm; for the general case of a knapsack constraint, Yu *et al.* [21] gave a single-pass $(1/3 - \varepsilon)$-approximation algorithm, both using $O(K \log(K)/\varepsilon)$ space.

We now state our contribution.

Theorem 1 *For the problem* (1)*,*

1. *there is a single-pass streaming algorithm with approximation ratio* $4/11 - \varepsilon \approx 0.363 - \varepsilon$.
2. *there is a multiple-pass streaming algorithm with approximation ratio* $2/5 - \varepsilon = 0.4 - \varepsilon$.

Both algorithms use $O(K \cdot \mathrm{poly}(\varepsilon^{-1})\mathrm{polylog}(K))$ *space.*

Our Technique We begin by a straightforward generalization of the algorithm of [2] for the special case of cardinality constraint (Section 2). This algorithm proceeds by adding a new item into the current set only if its marginal-ratio (its marginal return with respect to the current set divided by its size) exceeds a certain threshold. This algorithm performs well when all items in OPT are relatively small in size, where OPT is an optimal solution. However, in general, it only gives $(1/3 - \varepsilon)$-approximation. Note that this technique can be regarded as a variation of the one in [21]. To obtain better approximation ratio, we need new ideas.

The difficulty in improving this algorithm lies in the following case: A new arriving item that is relatively large in size, passes the marginal-ratio threshold, and is part of OPT, but its addition would cause the current set to exceed the capacity K. In this case, we are forced to throw it away, but in doing so, we are unable to bound the ratio of the function value of the current set against that of OPT properly.

We propose a branching procedure to overcome this issue. Roughly speaking, when the function value of the current set is large enough (depending on the parameters), we create a secondary set. We add an item to the secondary set only if it passes the marginal-ratio threshold (with respect to the original set) but its addition to the original set would violate the size constraint. In the end, whichever set achieves the higher value is returned. In a way, the secondary set serves as a "back-up" with enough space in case the original set does not have it, and this allows us to bound the ratio properly. Sections 3 and 4 are devoted to explaining this branching algorithm, which gives $(4/11 - \varepsilon)$-approximation with a single pass.

We note that the main bottleneck of the above singe-pass algorithm lies in the situation where there is a large item in OPT whose size exceeds $K/2$. In Section 5, we show that we can first focus on only

Algorithm 1

```
1: procedure MarginalRatioThresholding(α, v)                    ▷ α ∈ (0, 1], v ∈ ℝ+
2:     S := ∅.
3:     while item e is arriving do
4:         if f(e|S)/c(e) ≥ (αv − f(S))/(K − c(S)) and c(S + e) ≤ K then S := S + e.
5:     return S.
```

the large items (more specifically, those items whose size differ from the largest item in OPT by $(1+\varepsilon)$ factor) and choose $O(1)$ of them so that at least one of them, along with the rest of OPT (excluding the largest item in it), gives a good approximation to $f(\mathrm{OPT})$. Then in the next pass, we can apply a modified version of the original single-pass algorithm to collect small items. This multiple-pass algorithm gives a $(2/5-\varepsilon)$-approximation.

Related Work Maximizing a monotone submodular function subject to various constraints is a subject that has been extensively studied in the literature. We are unable to give a complete survey here and only highlight the most representative and relevant results. Besides a knapsack constraint or a cardinality constraint mentioned above, the problem has also been studied under (multiple) matroid constraint(s), p-system constraint, multiple knapsack constraints. See [4, 9, 13, 8, 15] and the references therein. In the streaming setting, other than the knapsack constraint that we have discussed before, there are also works considering a matroid constraint. Chakrabarti and Kale [5] gave 1/4-approximation; Chekuri *et al.* [7] gave the same ratio. Very recently, for the special case of partition matroid, Chan *et al.* [6] improved the ratio to 0.3178.

Notation For a subset $S \subseteq E$ and an element $e \in E$, we use the shorthand $S+e$ and $S-e$ to stand for $S \cup \{e\}$ and $S \setminus \{e\}$, respectively. For a function $f : 2^E \to \mathbb{R}$, we also use the shorthand $f(e)$ to stand for $f(\{e\})$. The *marginal return* of adding $e \in E$ with respect to $S \subseteq E$ is defined as $f(e \mid S) = f(S+e) - f(S)$. We frequently use the following, which is immediate from the diminishing marginal return property:

Proposition 2 *Let $f : 2^E \to \mathbb{R}_+$ be a monotone submodular function. For two subsets $S \subseteq T \subseteq E$, it holds that $f(T) \leq f(S) + \sum_{e \in T \setminus S} f(e \mid S)$.*

Due to the space limitation, the proofs of most lemmas and theorems are omitted, which can be found in the full version of this paper.

2 Single-Pass $(1/3-\varepsilon)$-Approximation Algorithm

In this section, we present a simple $(1/3-\varepsilon)$-approximation algorithm that generalizes the algorithm for a cardinality constraint in [2]. This algorithm will be incorporated into several other algorithms introduced later.

2.1 Thresholding Algorithm with Approximate Optimal Value

In this subsection, we present an algorithm MarginalRatioThresholding, which achieves (almost) 1/3-approximation given a (good) approximation v to $f(\mathrm{OPT})$ for an optimal solution OPT. This assumption is removed in Section 2.2.

Given a parameter $\alpha \in (0,1]$ and $v \in \mathbb{R}_+$, MarginalRatioThresholding attempts to add a new item $e \in E$ to the current set $S \subseteq E$ if its addition does not violate the knapsack constraint and e passes the *marginal-ratio threshold condition*, i.e.,

$$\frac{f(e \mid S)}{c(e)} \geq \frac{\alpha v - f(S)}{K - c(S)}. \tag{2}$$

Algorithm 2

```
1: procedure Singleton()
2:     S := ∅
3:     while item e is arriving do
4:         if f(e) > f(S) then S := {e}.
5:     return S.
```

The detailed description of MarginalRatioThresholding is given in Algorithm 1.

Throughout this subsection, we fix $\tilde{S} = \mathsf{MarginalRatioThresholding}(\alpha, v)$ as the output of the algorithm. Then, we have the following lemma.

Lemma 3 *The following hold:*

(1) *During the execution of the algorithm, the current set $S \subseteq E$ always satisfies $f(S) \geq \alpha v c(S)/K$. Moreover, if an item $e \in E$ passes the condition (2) with the current set S, then $f(S+e) \geq \alpha v c(S+e)/K$.*

(2) *If an item $e \in E$ fails the condition (2), i.e., $\frac{f(e|S)}{c(e)} < \frac{\alpha v - f(S)}{K - c(S)}$, then we have $f(e \mid \tilde{S}) < \alpha v c(e)/K$.*

An item $e \in \mathrm{OPT}$ is not added to $\tilde{S}$ if either e does not pass the condition (2), or its addition would cause the size of S to exceed the capacity K. We name the latter condition as follows:

Definition 4 *An item $e \in \mathrm{OPT}$ is called* bad *if e passes the condition (2) but the total size exceeds K when added, i.e., $f(e \mid S) \geq \frac{\alpha v - f(S)}{K - c(S)}$, $c(S+e) > K$ and $c(S) \leq K$, where S is the set we have just before e arrives.*

The following lemma says that, if there is no bad item, then we obtain a good approximation.

Lemma 5 *If $v \leq f(\mathrm{OPT})$ and there have been no bad item, then $f(\tilde{S}) \geq (1-\alpha)v$ holds.*

The following lemma says that, if we do not have a large item in OPT, then we can achieve (almost) 1/3-approximation.

Lemma 6 *If every item $e \in \mathrm{OPT}$ satisfies $c(e) \leq K/2$, then $f(\tilde{S}) \geq \min\{\alpha/2, 1-\alpha\}v$. In particular, $f(\tilde{S}) \geq v/3$ when $\alpha = 2/3$.*

Consider an algorithm Singleton, which takes the best singleton as shown in Algorithm 2. If some item $e \in \mathrm{OPT}$ has $c(e) > K/2$, then together with $\tilde{S}' = \mathsf{Singleton}()$, we have the same ratio of 1/3:

Theorem 7 *We have $\max\{f(\tilde{S}), f(\tilde{S}')\} \geq \min\{\alpha/2, 1-\alpha\}v$. The right-hand side is maximized to $v/3$ when $\alpha = 2/3$.*

Therefore, if we have $v \in \mathbb{R}_+$ with $v \leq f(\mathrm{OPT}) \leq (1+\varepsilon)v$, the algorithm that runs MarginalRatioThresholding$(2/3, v)$ and Singleton() in parallel and chooses the better output has the approximation ratio of $\frac{1}{3(1+\varepsilon)} \geq 1/3 - \varepsilon$. The space complexity of the algorithm is clearly $O(K)$.

2.2 Dynamic Updates

MarginalRatioThresholding requires a good approximation to $f(\mathrm{OPT})$. This requirement can be removed with dynamic updates in a similar way to [2]. We first observe that $\max_{e \in S} f(e) \leq f(\mathrm{OPT}) \leq K \max_{e \in S} f(e)$. So if we are given $m = \max_{e \in S} f(e)$ in advance, a value $v \in \mathbb{R}_+$ with $v \leq f(\mathrm{OPT}) \leq (1+\varepsilon)v$ for $\varepsilon \in (0, 1]$ exists in the guess set $\mathcal{I} = \{(1+\varepsilon)^i \mid m \leq (1+\varepsilon)^i \leq Km, i \in \mathbb{Z}_+\}$. Then, we can run MarginalRatioThresholding for each $v \in \mathcal{I}$ in parallel and choose the best output. As the size of $\mathcal{I}$ is $O(\log K/\varepsilon)$, the total space complexity is $O(K \log K/\varepsilon)$.

To get rid of the assumption that we are given m in advance, we consider an algorithm, called DynamicMRT, which dynamically updates m to determine the range of guessed optimal values. More specifically, it keeps the (tentative) maximum value $\max f(e)$, where the maximum is taken over the items e arrived so far, and keeps the approximations v in the interval between m and Km/α. The details are provided in Algorithm 3.

Algorithm 3

procedure DynamicMRT(ε, α) ▷ $\varepsilon, \alpha \in (0, 1]$
  $\mathcal{V} := \{(1+\varepsilon)^i \mid i \in \mathbb{Z}_+\}$.
  For each $v \in \mathcal{V}$, set $S_v := \emptyset$.
  while item e is arriving **do**
    $m := \max\{m, f(e)\}$
    $\mathcal{I} := \{v \in \mathcal{V} \mid m \leq v \leq Km/\alpha\}$.
    Delete S_v for each $v \notin \mathcal{I}$.
    for each $v \in \mathcal{I}$ **do**
      if $\frac{f(e|S_v)}{c(e)} \geq \frac{\alpha v - f(S_v)}{K - c(S_v)}$ and $c(S_v + e) \leq K$ **then** $S_v := S_v + e$.
  return S_v for $v \in \mathcal{I}$ that maximizes $f(S_v)$.

Theorem 8 *For $\varepsilon \in (0, 1]$, the algorithm that runs* DynamicMRT($\varepsilon, 2/3$) *and* Singleton() *in parallel and outputs the better output is a $(1/3 - \varepsilon)$-approximation streaming algorithm with a single pass for the problem* (1)*. The space complexity of the algorithm is $O(K \log K/\varepsilon)$.*

3 Improved Single-Pass Algorithm for Small-Size Items

Let OPT $= \{o_1, o_2, \ldots, o_\ell\}$ be an optimal solution with $c(o_1) \geq c(o_2) \geq \cdots \geq c(o_\ell)$. The main goal of this section is achieving $(2/5 - \varepsilon)$-approximation, assuming that $c(o_1) \leq K/2$. The case with $c(o_1) > K/2$ will be discussed in Section 4.

3.1 Branching Framework with Approximate Optimal Value

We here provide a framework of a branching algorithm BranchingMRT as Algorithm 4. This will be used with different parameters in Section 3.2.

Let v and c_1 be (good) approximations to $f(\mathrm{OPT})$ and $c(o_1)/K$, respectively, and let $b \leq 1/2$ be a parameter. The value c_1 is supposed to satisfy $c_1 \leq c(o_1)/K \leq (1+\varepsilon)c_1$, and hence we ignore items $e \in E$ with $c(e) > \min\{(1+\varepsilon)c_1, 1/2\}K$. The basic idea of BranchingMRT is to take only items with large marginal ratios, similarly to MarginalRatioThresholding. The difference is that, once $f(S)$ exceeds a threshold λ, where $\lambda = \frac{1}{2}\alpha\,(1-b)\,v$, we store either the current set S or the latest added item as S'. This guarantees that $f(S') \geq \lambda$ and $c(S') \leq (1-b)K$, which means that S' has a large function value and sufficient room to add more elements. We call the process of constructing S' *branching*. We continue to add items with large marginal ratios to the current set S, and if we cannot add an item to S because it exceeds the capacity, we try to add the item to S'. Note that the set S', after branching, can have at most one extra item; but this extra item can be replaced if a better candidate comes along (See line 14–15).

Remark that the sequence of sets S in BranchingMRT is identical to that in MarginalRatioThresholding. Hence, we do not need to run MarginalRatioThresholding in parallel to this algorithm. We say that an item $e \in \mathrm{OPT}$ is *bad* if it is bad in the sense of MarginalRatioThresholding, i.e., it satisfies the condition in Definition 4. We have the following two lemmas.

Lemma 9 *For a bad item e with $c(e) \leq bK$, let S_e be the set just before e arrives in Algorithm 4. Then $f(S_e) \geq \lambda$ holds. Thus branching has happened before e arrives.*

Lemma 10 *It holds that $f(S'_0) \geq \lambda$ and $c(S'_0) \leq (1-b)K$.*

Let $\tilde{S}$ and $\tilde{S}'$ be the final two sets computed by BranchingMRT. Note that we can regard $\tilde{S}$ as the output of MarginalRatioThresholding and $\tilde{S}'$ as the final set obtained by adding at most one item to S'_0.

Observe that the number of bad items depends on the parameter α. As we will show in Section 3.2, by choosing a suitable α, if we have more than two bad items, then the size of $\tilde{S}$ is large enough, implying that $f(\tilde{S})$ is already good for approximation (due to Lemma 3 (1)). Therefore, in the following, we just concentrate on the case when we have at most two bad items.

Algorithm 4

procedure BranchingMRT($\varepsilon, \alpha, v, c_1, b$) ▷ $\varepsilon, \alpha \in (0,1]$, $v \in \mathbb{R}_+$, and $c_1, b \in [0, 1/2]$
  $S := \emptyset$.
  $\lambda := \frac{1}{2}\alpha(1-b)v$.
  while item e is arriving **do**
    Delete e with $c(e) > \min\{(1+\varepsilon)c_1, 1/2\}K$.
    if $\frac{f(e|S)}{c(e)} \geq \frac{\alpha v - f(S)}{K - c(S)}$ and $c(S+e) \leq K$ **then** $S := S + e$.
    if $f(S) \geq \lambda$ **then break** // leave the While loop.
  Let $\hat{e}$ be the latest added item in S.
  if $c(S) \geq (1-b)K$ **then** $S'_0 := \{\hat{e}\}$ **else** $S'_0 := S$.
  $S' := S'_0$.
  while item e is arriving **do**
    Delete e with $c(e) > \min\{(1+\varepsilon)c_1, 1/2\}K$.
    if $\frac{f(e|S)}{c(e)} \geq \frac{\alpha v - f(S)}{K - c(S)}$ and $c(S+e) \leq K$ **then** $S := S + e$.
    if $\frac{f(e|S)}{c(e)} \geq \frac{\alpha v - f(S)}{K - c(S)}$ and $c(S+e) > K$ **then**
      if $f(S') < f(S'_0 + e)$ **then** $S' := S'_0 + e$.
  return S or S' whichever has the larger function value.

Lemma 11 *Let α be a number in $(0,1]$, and suppose that we have only one bad item o_{b}. If $v \leq f(\mathrm{OPT})$ and $b \in [c(o_{\mathrm{b}})/K, (1+\varepsilon)c(o_{\mathrm{b}})/K]$, then it holds that*

$$\max\{f(\tilde{S}), f(\tilde{S}')\} \geq \left(\frac{1}{2}\left(1 - \alpha\frac{K - c(o_{\mathrm{b}})}{2K}\right) - O(\varepsilon)\right)v.$$

Lemma 12 *Let α be a number in $(0,1]$, and suppose that we have exactly two bad items o_{b} and o_{m} with $c(o_{\mathrm{b}}) \geq c(o_{\mathrm{m}})$. If $v \leq f(\mathrm{OPT})$ and $b \in [c(o_{\mathrm{b}})/K, (1+\varepsilon)c(o_{\mathrm{b}})/K]$, then it holds that*

$$\max\{f(\tilde{S}), f(\tilde{S}')\} \geq \left(\frac{1}{3}\left(1 + \alpha\frac{c(o_{\mathrm{m}})}{K}\right) - O(\varepsilon)\right)v.$$

3.2 Algorithms with Guessing Large Items

We now use BranchingMRT to obtain a better approximation ratio. In the new algorithm, we guess the sizes of a few large items in an optimal solution OPT, and then use them to determine the parameter α.

We first remark that, when $|\mathrm{OPT}| \leq 2$, we can easily obtain a $1/2$-approximate solution with a single pass. In fact, since $f(\mathrm{OPT}) \leq \sum_{i=1}^{\ell} f(o_i)$ where $\ell = |\mathrm{OPT}|$, at least one of o_i's satisfies $f(o_i) \geq f(\mathrm{OPT})/\ell$, and hence Singleton returns a $1/2$-approximate solution when $\ell \leq 2$. Thus, in what follows, we may assume that $|\mathrm{OPT}| \geq 3$.

We start with the case that we have guessed the largest two sizes $c(o_1)$ and $c(o_2)$ in OPT. Then, we have the following:

Lemma 13 *Let $\varepsilon \in (0,1]$, and suppose that $v \leq f(\mathrm{OPT})$ and $c_i \leq c(o_i)/K \leq (1+\varepsilon)c_i$ for $i \in \{1,2\}$. Then, $\tilde{S}' = \mathsf{BranchingMRT}(\varepsilon, \alpha, v, c_1, b)$ with $\alpha = 1/(2-c_2)$ or $2/(5-4c_2-c_1)$ and $b = \min\{(1+\varepsilon)c_1, 1/2\}$ satisfies*

$$f(\tilde{S}') \geq \left(\min\left\{\frac{1-c_2}{2-c_2}, \frac{2(1-c_2)}{5-4c_2-c_1}\right\} - O(\varepsilon)\right)v.$$

Note that the approximation ratio achieved in Lemma 13 becomes $1/3 - O(\varepsilon)$ when, for example, $c_1 = c_2 = 1/2$. Hence, the above lemma does not show any improvement over Theorem 7 in the worst case. Thus, we next consider the case that we have guessed the largest three sizes $c(o_1)$, $c(o_2)$, and $c(o_3)$ in OPT.

Lemma 14 *Let $\varepsilon \in (0,1]$, and suppose that $v \leq f(\mathrm{OPT})$ and $c_i \leq c(o_i)/K \leq (1+\varepsilon)c_i$ for $i \in \{1,2,3\}$. Then the better output $\tilde{S}'$ of $\mathsf{BranchingMRT}(\varepsilon, \alpha, v, c_1, b_1)$ and $\mathsf{BranchingMRT}(\varepsilon, \alpha, v, c_1, b_2)$ with $\alpha =$*

$1/(2-c_3)$ *or* $2/(c_2+3)$, $b_1 = \min\{(1+\varepsilon)c_1, 1/2\}$, *and* $b_2 = \min\{(1+\varepsilon)c_2, 1/2\}$ *satisfies*

$$f(\tilde{S}') \geq \left(\min\left\{\frac{1-c_3}{2-c_3}, \frac{c_2+1}{c_2+3}\right\} - O(\varepsilon)\right) v.$$

We now see that we get an approximation ratio of $2/5 - O(\varepsilon)$ by combining the above two lemmas.

Theorem 15 *Let $\varepsilon \in (0,1]$ and suppose that $v \leq f(\mathrm{OPT}) \leq (1+\varepsilon)v$ and $c_i \leq c(o_i)/K \leq (1+\varepsilon)c_i$ for $i \in \{1,2,3\}$. If $c(o_1) \leq K/2$, then we can obtain a $(2/5 - O(\varepsilon))$-approximate solution with a single pass.*

Proof: We run the two algorithms with the optimal α shown in Lemmas 13 and 14 in parallel. Let $\tilde{S}$ be the output with the better function value. Then, we have $f(\tilde{S}) \geq \beta v$, where

$$\beta = \max\left\{\min\left\{\frac{1-c_2}{2-c_2}, \frac{2(1-c_2)}{5-4c_2-c_1}\right\}, \min\left\{\frac{1-c_3}{2-c_3}, \frac{c_2+1}{c_2+3}\right\}\right\} - O(\varepsilon).$$

We can confirm that the first term is at least $2/5$, and thus $\tilde{S}$ is a $(2/5 - O(\varepsilon))$-approximate solution. □

To eliminate the assumption that we are given v, we can use the same technique as in Theorem 8. Similarly to Theorem 8, we can design a dynamic-update version of BranchingMRT by keeping the interval that contains the optimal value. The number of streams for guessing v is $O(\log K/\varepsilon)$. We also guess c_i for $i \in \{1,2,3\}$ from $\{(1+\varepsilon)^j \mid j \in \mathbb{Z}_+\}$. As $1 \leq c(o_i) \leq K/2$, the number of guessing for c_i is $O(\log K/\varepsilon)$. Therefore, there are $O((\log K/\varepsilon)^4)$ streams in total. To summarize, we obtain the following:

Theorem 16 *Suppose that $c(o_1) \leq K/2$. The algorithm that runs* DynamicBranchingMRT *and* Singleton *in parallel and takes the better output is a $(2/5-\varepsilon)$-approximation streaming algorithm with a single pass for the problem* (1)*. The space complexity of the algorithm is $O(K(\log K/\varepsilon)^4)$.*

4 Single-Pass $(4/11-\varepsilon)$-Approximation Algorithm

In this section, we consider the case that $c(o_1)$ is larger than $K/2$. For the purpose, we consider the problem of finding a set S of items that maximizes $f(S)$ subject to the constraint that the total size is at most pK, for a given number $p \geq 2$. We say that a set S of items is a (p,α)-*approximate solution* if $c(S) \leq pK$ and $f(S) \geq \alpha f(\mathrm{OPT})$, where OPT is an optimal solution of the original instance.

Theorem 17 *For a number $p \geq 2$, there is a $\left(p, \frac{2p}{2p+3} - \varepsilon\right)$-approximation streaming algorithm with a single pass for the problem* (1)*. In particular, when $p = 2$, it admits $(2, 4/7-\varepsilon)$-approximation. The space complexity of the algorithm is $O(K(\log K/\varepsilon)^3)$.*

The basic framework of the algorithm is the same as in Section 3; we design a thresholding algorithm and a branching algorithm, where the parameters are different and the analysis is simpler.

Using Theorem 17, we can design a $(4/11-\varepsilon)$-approximation streaming algorithm for an instance having a large item.

Theorem 18 *For the problem* (1)*, there exists a $(4/11-\varepsilon)$-approximation streaming algorithm with a single pass. The space complexity of the algorithm is $O(K(\log K/\varepsilon)^4)$.*

Proof: Let o_1 be an item in OPT with the maximum size. If $c(o_1) \leq K/2$, then Theorem 16 gives a $(2/5-O(\varepsilon))$-approximate solution, and thus we may assume that $c(o_1) > K/2$. Note that there exists only one item whose size is more than $K/2$. Let β be the target approximation ratio which will be determined later. We may assume that $f(o_1) < \beta v$, where $v = f(\mathrm{OPT})$, otherwise Singleton (Algorithm 2) gives β-approximation. Then, we see $f(\mathrm{OPT} - o_1) > (1-\beta)f(\mathrm{OPT})$ and $c(\mathrm{OPT} - o_1) < K/2$. Consider

maximizing $f(S)$ subject to $c(S) \leq K/2$ in the set $\{e \in E \mid c(e) \leq K/2\}$. The optimal value is at least $f(\mathrm{OPT} - o_1) > (1-\beta)f(\mathrm{OPT})$. We now apply Theorem 17 with $p = 2$ to this problem. Then, the output $\tilde{S}$ has size at most K, and moreover, we have

$$f(\tilde{S}) \geq \left(\frac{4}{7} - O(\varepsilon)\right)(1-\beta)f(\mathrm{OPT}).$$

Thus, we obtain $\min\{\beta, (\frac{4}{7} - O(\varepsilon))(1-\beta)\}$-approximation. This approximation ratio is maximized to $4/11$ when $\beta = 4/11$. □

5 Multiple-Pass Streaming Algorithm

In this section, we provide a multiple-pass streaming pass algorithm with approximation ratio $2/5 - \varepsilon$.

We first consider a generalization of the original problem. Let $E_R \subseteq E$ be a subset of the ground set E. For ease of presentation, we will call E_R the *red* items. Consider the problem defined below:

$$\text{maximize} \quad f(S) \quad \text{subject to} \quad c(S) \leq K, |S \cap E_R| \leq 1. \tag{3}$$

In the following, we show that, given $\varepsilon \in (0, 1]$, an approximation v to $f(\mathrm{OPT})$ with $v \leq f(\mathrm{OPT}) \leq (1+\varepsilon)v$, and an approximation θ to $f(o_{\mathrm{r}})$ for the unique item o_{r} in $\mathrm{OPT} \cap E_R$, we can choose $O(1)$ of the red items so that one of them $e \in E_R$ satisfies that $f(\mathrm{OPT} - o_{\mathrm{r}} + e) \geq (\Gamma(\theta) - O(\varepsilon))v$, where $\Gamma(\cdot)$ is a piecewise linear function lower-bounded by $2/3$. For technical reasons, we will choose θ to be one of the geometric series $(1+\varepsilon)^i/2$ for $i \in \mathbb{Z}$.

Theorem 19 *Suppose that we are given $\varepsilon \in (0, 1]$, $v \in \mathbb{R}_+$ with $v \leq f(\mathrm{OPT}) \leq (1+\varepsilon)v$, and $\theta \in \mathbb{R}_+$ with the following property: (1) if $\theta \leq 1/2$, $\theta v/(1+\varepsilon) \leq f(o_{\mathrm{r}}) \leq \theta v$, and (2) if $\theta \geq 1/2$, $\theta v \leq f(o_{\mathrm{r}}) \leq (1+\varepsilon)\theta v \leq v$. Then, there is a single-pass streaming algorithm that chooses a constant number of red items in E_R so that one item e of them satisfies that $f(\mathrm{OPT} - o_{\mathrm{r}} + e) \geq v(\Gamma(\theta) - O(\varepsilon))$, where*

- *$\Gamma(\theta)$ is defined as the following function when $\theta \in (0, 1/2)$:*

$$\Gamma(\theta) = \max\Big\{\frac{t(t+3)}{(t+1)(t+2)} - \frac{t-1}{t+1}\theta \mid t \in \mathbb{Z}_+, t > \frac{1}{\theta} - 2\Big\}.$$

- *$\Gamma(\theta) = 2/3$ when $\theta \in [1/2, 2/3)$,*
- *$\Gamma(\theta) = \theta$ when $\theta \in [2/3, 1]$.*

We next show that when $c(o_1) \geq K/2$, we can use multiple passes to get a $(2/5 - \varepsilon)$-approximation for the problem (1). Let $\mathrm{OPT} = \{o_1, o_2, \ldots, o_\ell\}$ be an optimal solution with $c(o_1) \geq c(o_2) \geq \cdots \geq c(o_\ell)$. Suppose that $c_1 \in \mathbb{R}_+$ satisfies $1/2 \leq c_1/(1+\varepsilon) \leq c(o_1)/K \leq c_1$.

We observe the following claims.

Claim 20 *When $c(o_1) \geq K/2$, we may assume that $\frac{3}{10}f(\mathrm{OPT}) < f(o_1) < \frac{2}{5}f(\mathrm{OPT})$.*

Claim 21 *We may assume that $c(o_1) \leq (1+\varepsilon)\frac{2}{3}K$.*

We use the first pass to estimate $f(\mathrm{OPT})$ as follows. For an error parameter $\varepsilon \in (0, 1]$, perform the single-pass algorithm in Theorem 8 to get a $(1/3 - \varepsilon)$-approximate solution $S \subseteq E$, which can be used to upper bound the value of $f(\mathrm{OPT})$, that is, $f(S) \leq f(\mathrm{OPT}) \leq (3+\varepsilon)f(S)$. We then find the geometric series to guess its exact value. Thus, we may assume that we are given the value v with $v \leq f(\mathrm{OPT}) \leq (1+\varepsilon)v$.

Below we show how to obtain a solution of value at least $(2/5 - O(\varepsilon))v$, using two more passes. Before we start, we introduce a slightly modified versions of the algorithms presented in Section 2; it will be used as a subroutine.

Lemma 22 *Consider the problem* (1) *with the knapsack capacity* K'*. Let* $h \in \mathbb{R}_+$*, and suppose that Algorithms 1 and 2 are modified as follows:*

- *(At Line 4 in Algorithm 1) A new item* e *is added into the current set* S *only if* $\frac{f(e|S)}{c(e)} \geq \frac{\alpha v - f(S)}{hK' - c(S)}$ *and* $c(S+e) \leq hK'$*.*
- *(At Line 4 in Algorithm 2) A new item* e *is taken into account only if* $c(e) \leq hK'$*.*

Then, the best returned set $\tilde{S}$ *of the two algorithms with* $\alpha = \frac{2h}{h+2}$ *satisfies that* $c(\tilde{S}) \leq hK'$ *and* $f(\tilde{S}) \geq \frac{h}{h+2}v$*. Moreover, we can obtain a* $\left(\frac{h}{h+2} - O(\varepsilon)\right)$*-approximate solution with the dynamic update technique.*

Let all items $e \in E$ whose sizes $c(e)$ satisfy $c_1/(1+\varepsilon) \leq c(e)/K \leq c_1$ be the red items. By Theorem 19, we can select a set S of the red items so that one of them guarantees $f(\mathrm{OPT} - o_1 + e) \geq (\Gamma(\theta) - O(\varepsilon))v$, where θ satisfies the condition in Theorem 19. Note that any $e \in S$ satisfies $f(e) \geq \theta v/(1+\varepsilon)$. Also, by Claim 20, we see $\frac{3}{10}v < \theta < \frac{2}{5}(1+\varepsilon)v$.

In the next pass, for each $e \in S$, define a new monotone submodular function $g_e(\cdot) = f(\cdot \mid e)$ and apply the marginal-ratio thresholding algorithm (Lemma 22) with regard to function g_e, where $h = \frac{1-c_1}{1-(c_1/(1+\varepsilon))}$ and $K' = (1 - (c_1/(1+\varepsilon))K$.

The returned solution has size at most K, since $c(S_e) \leq hK' = (1-c_1)K$ by Lemma 22. The next theorem summarizes our results in this section.

Theorem 23 *Suppose that* $c(o_1) > K/2$*. There exists an algorithm that uses* MultiPassKnapsack *as a subroutine so that it returns* $(2/5 - \varepsilon)$*-approximation with 3 passes for the problem* (1)*. The space complexity of the algorithm is* $O(K(\log K/\varepsilon)^2)$*.*

References

[1] N. Alon, I. Gamzu, and M. Tennenholtz. Optimizing budget allocation among channels and influencers. In *Proceedings of the 21st International Conference on World Wide Web (WWW)*, pages 381–388, 2012.

[2] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 671–680, 2014.

[3] A. Badanidiyuru and J. Vondrák. Fast algorithms for maximizing submodular functions. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1497–1514, 2013.

[4] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[5] A. Chakrabarti and S. Kale. Submodular maximization meets streaming: matchings, matroids, and more. *Mathematical Programming*, 154(1-2):225–247, 2015.

[6] T.-H. H. Chan, Z. Huang, S. H.-C. Jiang, N. Kang, and Z. G. Tang. Online submodular maximization with free disposal: Randomization beats for partition matroids online. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1204–1223, 2017.

[7] C. Chekuri, S. Gupta, and K. Quanrud. Streaming algorithms for submodular function maximization. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 9134, pages 318–330, 2015.

[8] C. Chekuri, J. Vondrák, and R. Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.

[9] Y. Filmus and J. Ward. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. *SIAM Journal on Computing*, 43(2):514–542, 2014.

[10] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions ii. *Mathematical Programming Study*, 8:73–87, 1978.

[11] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 137–146, 2003.

[12] A. Krause, A. P. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[13] A. Kulik, H. Shachnai, and T. Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 545–554, 2013.

[14] J. Lee. *Maximum Entropy Sampling*, volume 3 of *Encyclopedia of Environmetrics*, pages 1229–1234. John Wiley & Sons, Ltd., 2006.

[15] J. Lee, M. Sviridenko, and J. Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research*, 35(4):795–806, 2010.

[16] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 912–920, 2010.

[17] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 510–520, 2011.

[18] T. Soma, N. Kakimura, K. Inaba, and K. Kawarabayashi. Optimal budget allocation: Theoretical guarantee and efficient algorithm. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 351–359, 2014.

[19] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

[20] L. Wolsey. Maximising real-valued submodular functions: primal and dual heuristics for location problems. *Mathematics of Operations Research*, 1982.

[21] Q. Yu, E. L. Xu, and S. Cui. Streaming algorithms for news and scientific literature recommendation: Submodular maximization with a d-knapsack constraint. *IEEE Global Conference on Signal and Information Processing*, 2016.

On the chip-firing halting problem for undirected multigraphs

Bálint Hujter[1]

MTA-ELTE Egerváry Research Group,
Eötvös Loránd University
Pázmány Péter sétány 1/C, H-1117 Budapest,
Hungary
hujterb@cs.elte.hu

Abstract: We address the chip-firing halting problem for undirected multigraphs. We give a polynomial algorithm for the special case when the number of chips in the distribution is equal to the sum of the edge multiplicities of the graph. The key part of our algorithm uses convex cost flow optimization to give an efficient algorithmic proof of a theorem of An, Baker, Kuperberg and Shokrieh; improving a previous algorithm of Backman.

Keywords: chip-firing games; computational complexity

1 Introduction

Chip-firing is a solitary game on graphs defined by Björner, Lovász and Shor [5, 4]. Each node contains a pile of chips. A legal move is to choose a node with at least as many chips as its out-degree and let it send a chip along each outgoing edge.

Björner, Lovász and Shor proved that given an initial chip-distribution, either every chip-firing game can be continued indefinitely, or every game terminates after finitely many steps. Tardos [16] proved that on a simple undirected graph on n nodes, a finite chip-firing game terminates in $O(n^4)$ firings. On the other hand, for digraphs the number of firings in a terminating chip-firing game is not necessarily polynomial [8]. It is not well-known that if we consider an undirected multigraph given by an adjacency matrix, then the number of firings in a terminating game is pseudo-polynomial but not necessarily polynomial in the input size (see Example 13). Therefore, it is a nontrivial question whether there exists a polynomial algorithm deciding whether a chip-distribution x is terminating or not. This is called the *chip-firing halting problem.*

	Undirected	Directed (Strongly Connected)
Simple Graphs	in **P** [16]	in **NP** [10]
Multigraphs	in **NP**∩**co-NP** [13]	**NP**-complete [10]

Table 1: Overview of the main complexity results concerning the halting problem.

In [10], Farrell and Levine proved that the halting problem for (strongly connected) directed multigraphs is **NP**-complete and noted that the question is open for undirected multigraphs and directed simple graphs either.

In this paper, we focus on undirected multigraphs and aim the following conjecture.

[1]Research is supported by the Hungarian Scientic Research Fund - OTKA K109240.

Conjecture 1 *The chip-firing halting problem for undirected multigraphs is in* **P**.

Several reasons support this conjecture. First, the author with V. Kiss and L. Tóthmérész recently noticed in [13] that the problem is in **NP**∩**co-NP**, even in the more general case of Eulerian directed multigraphs (for completeness, we include a proof in Section 3.1). Second, the closely related problem of reachability of chip-distributions [4] is in **P** [13]. Third, as the main result of this paper, we prove the following special case of the conjecture:

Theorem 2 *There is a polynomial algorithm deciding the halting problem for undirected multigraphs, in the special case of distributions of exactly M chips, where M denotes the sum of the edge multiplicities of the multigraph.*

Note that a game with less than M chips is always terminating (by applying [5, Theorem 3.3.] to multigraphs), so our algorithm concerns the minimal non-trivial case. The algorithm is similar to Backman's Algorithm 7.7 of [2] which finds an orientation representing an arbitrary chip-distribution of M chips, proving [1, Theorem 4.10.]. Backman's algorithm uses max-flow-min-cut computations, but the number of MFMC subroutines needed is pseudo-polynomial in the multigraph case. We obtain a polynomial algorithm by compressing the main part of the algorithm into one convex cost flow optimization problem.

Chip-firing games have a strong connection to the graph divisor theory introduced by Baker and Norine in [3]. Section 3.3 briefly describes how the halting problem is represented in graph divisor theory.

2 Preliminaries

2.1 Multigraphs

A *multigraph* is an undirected graph $G = (V, E)$ with an edge multiplicity function $m : E \to \mathbb{N}$. As G is undirected, $vw \in E$ iff $wv \in E$ and $m_{vw} = m_{wv}$.

We use the notations $d(v) = \sum_{w:vw\in E} m_{vw}$ and $M = \sum_{e\in E} m(e)$. Note that $|E| = O(|V|^2)$, but M is not necessarily polynomial in $|V|$.

When we give a multigraph as an input to an algorithm, we always encode it by its adjacency matrix. As the adjacency matrix can be encoded in $O(|V|^2 \log M)$ bits. Hence the size of the input is not increased by the values of the edge multiplicities, just the logarithms of it.

Remark 3 *We are only considering undirected graphs. Note that some of the following results remains true or have an equivalent version for directed graphs, but the main results of the paper only concern undirected graphs.*

The *Laplacian matrix* of a multigraph G is the following matrix $L \in \mathbb{Z}^{V\times V}$:

$$L(v,w) = \begin{cases} -d(v) & \text{if } v = w; \\ m_{vw} & \text{if } v \neq w \text{ and } vw \in E; \\ 0 & \text{if } v \neq w \text{ and } vw \notin E. \end{cases}$$

We define orientations of multigraphs the following way: if we have an edge vw with multiplicity m_{vw}, then we consider it as m_{vw} pieces of single edges, and one may orient each single edge independently from the others. An orientation of G is described by a number $0 \leq \overrightarrow{m}_{vw} \leq m_{vw}$ for each $vw \in E$, meaning that $\overrightarrow{m}_{vw}$ pieces of edges are oriented from v to w and $\overrightarrow{m}_{wv} = m_{vw} - \overrightarrow{m}_{vw}$ pieces are oriented from w to v.

We call an edge vw *consistently oriented* if $\overrightarrow{m}_{wv} \in \{0, m_{vw}\}$, and call an orientation *consistent*, if all the edges are consistently oriented. We use the following notation for the in-degree of a node v: $d^-_{\mathcal{O}}(v) = \sum_{w:wv\in E} \overrightarrow{m}_{wv}$. Node v is called a *sink* if $d^-_{\mathcal{O}}(v) = d(v)$ and a *source* if $d^-_{\mathcal{O}}(v) = 0$.

We introduce notations for some vectors of $\mathbb{Z}^V$. $\mathbf{1}$ stands for the vector with all 1s; $\mathbf{1}_v$ stands for the vector with 1 on a single node v and 0 elsewhere. $\mathbf{d}$ denotes the vector of $d(v)$s (degrees) in G and $\mathbf{d}^-_{\mathcal{O}}$ denotes the vector of $d^-_{\mathcal{O}}(v)$s (indegrees) of the orientation $\mathcal{O}$ of G.

2.2 Chip-firing games on multigraphs

Chip-firing games of Björner, Lovász and Shor [5] have a straightforward generalization for multigraphs. We consider a multigraph G with a pile of chips on each of its nodes. A position of the game, called a *chip-distribution* (or just distribution) is described by a vector $x \in \mathbb{Z}^V$, where $x(v)$ is interpreted as the number of chips on node $v \in V$. We denote the set of all chip-distributions on G by $\mathrm{Chip}(G)$.

The basic move of the game is *firing* a node. Firing node v means that for every $vw \in E$, node v passes m_{vw} chips to w. In other words, firing a node v means taking the new chip-distribution $x + L\mathbf{1}_v$ instead of x.

A node $v \in V$ is active with respect to a chip-distribution x if $x(v) \geq d(v)$. The firing of a node $v \in V$ is *legal*, if v was active before the firing (i.e. v has a nonnegative amount of chips after the firing). A *legal game* is a sequence of distributions in which every distribution is obtained from the previous one by a legal firing. A legal game terminates if it arrives at *stable* distribution, which is a chip-distribution without any active nodes. For a legal game, let us call the vector $f \in \mathbb{Z}^V$, where $f(v)$ equals the number of times v has been fired, the *firing vector* of the game.

We use the notation $\deg(x) = \sum_{v \in V} x(v)$ for any $x \in \mathrm{Chip}(G)$. It is easy to check that $\deg(x)$ is invariant during a chip-firing game.

The following theorem of Björner, Lovász and Shor describes a very important "Abelian" property of the chip-firing game.

Theorem 4 *[5, Remark 2.4] From a given initial chip-distribution, either every legal game can be continued indefinitely, or every legal game terminates after finitely many steps. The firing vector of every maximal legal game is the same.*

Based on this fact, we call a distribution x *terminating* if a legal game (hence, all legal games) started from x terminates, and we call x *non-terminating* otherwise.

2.3 Sink-reversal games

Variants of sink-reversal games have appeared in many forms in literature: [5, 9, 12] etc. Here we give a short summary of some basic results needed in this paper.

Let G be a multigraph with an orientation $\mathcal{O}$. The *sink-reversal game* on G is defined as follows. The basic move of the game is *reversing a sink*, which means that if a node v is a sink then we may reverse the orientation of all edges incident to v (this way v becomes a source in the newly obtained orientation). A game terminates if there are no sink nodes in the actual orientation.

Let $\mathcal{O}$ be any orientation of G and consider the in-degree vector $\mathbf{d}^-_{\mathcal{O}}$ as a chip-distribution. Notice that we can fire node v if and only if v is a sink in $\mathcal{O}$. After firing v, the resulting chip-distribution is the in-degree vector $\mathbf{d}^-_{\mathcal{O}'}$ of orientation $\mathcal{O}'$ obtained by sink-reversing v in $\mathcal{O}$. These observations imply the following claim.

Claim 5 *The chip-firing game started from $\mathbf{d}^-_{\mathcal{O}}$ has the same dynamics as the sink-reversal game started from $\mathcal{O}$: both are terminating or both are non-terminating. For any node v, the number of sink-reversals at v equals the number of firings at v.*

By Theorem 4, we get that from a given initial orientation, either every sink-reversal game can be continued indefinitely, or every legal game terminates after finitely many steps. The advantage of sink-reversal games is that we have a nice characterization of terminating games.

Proposition 6 *A sink-reversal game started from orientation $\mathcal{O}$ of G is terminating if and only if $\mathcal{O}$ contains a directed cycle.*

The proof of this Proposition can be an exercise for the reader. We included a detailed proof in the Appendix. Using the characterization it is also easy to decide algorithmically whether a sink-reversal game is terminating or not:

Proposition 7 *There is an algorithm running in $O(|E|)$ time for the following problem: given an orientation $\mathcal{O}$ of multigraph G, decide whether the sink-reversal game $\mathcal{O}$ is terminating or non-terminating.*

PROOF: By Proposition 6, we only need to check whether $\mathcal{O}$ contains a directed cycle or not, which can be done by a depth first search. If there is any edge which is not consistently oriented (i.e. $0 < \overrightarrow{m}_{vw} < m_{vw}$), then it gives rise to a directed cycle. If G is consistently oriented, it is enough to consider one copy of the consistently oriented parallel edges. Hence the running time is $O(|E|)$. □

2.4 Convex cost flows

A detailed description of convex cost flow problems and solution techniques can be found in Section 14 of [17]. Here we only give a short summary. We expect that the reader is familiar with the standard notations, theorems and techniques of minimum cost flow problems with linear cost functions (see for example [17, Chapters 9-11.], [18, Chapter 3.6.] or [19, Chapter 12.]).

In the general convex cost flow problem a directed network $\mathcal{N}$ is given with:

- node set V and arc set A;
- an *excess* $b(v) \in \mathbb{R}$ for any $v \in V$ such that $\sum_{v \in V} b(v) = 0$;
- a *capacity* $u_{vw} \in \mathbb{R}^+$ and a *convex cost function* $c_{vw} : [0, u_{vw}] \to \mathbb{R}^+$ for all $\overrightarrow{vw} \in A$ (c is convex in the common sense of convexity).

$f : A \to \mathbb{R}^+$ is a called a *flow* in $\mathcal{N}$ iff it satisfies the following two conditions:

$$\sum_{w:\overrightarrow{vw} \in A} f_{vw} - \sum_{w:\overrightarrow{wv} \in A} f_{wv} = b(v) \quad \text{for all} \quad v \in V \tag{1}$$

$$0 \le f_{vw} \le u_{vw} \quad \text{for all} \quad \overrightarrow{vw} \in A. \tag{2}$$

Our goal is to find an optimal flow in the following sense:

$$\sum_{\overrightarrow{vw} \in A} c_{vw}(f_{vw}) \to \min. \tag{3}$$

An important special case of the general convex cost flow problem is when c_{vw} is a piecewise linear integral convex cost function.

Definition 8 *Let $u \in \mathbb{N}$. Then $c : [0, u] \to \mathbb{R}^+$ is a* piecewise linear integral convex cost function *iff c is convex, $c(k) \in \mathbb{Z}$ for all $k \in \mathbb{Z}$ and $c(x)$ is linear between any pair of consecutive integers.*

In the case of piecewise linear integral convex cost functions, the convex cost flow problem can be transformed to a linear cost flow problem in the following way: we replace each arc $\overrightarrow{vw}$ by u_{vw} pieces of arcs, each with capacity 1 and the kth one with cost $c_{vw}(k) - c_{vw}(k-1)$ for all $k \in \{1, \dots, u_{vw}\}$, we call this new network $\mathcal{N}'$. Since a minimal cost flow in $\mathcal{N}'$ does not use an arc vw unless all arcs from v to w with a lower cost are saturated, there is a natural correspondence between min cost flows in $\mathcal{N}$ and $\mathcal{N}'$.

As network $\mathcal{N}'$ has linear cost functions, optimal flows in $\mathcal{N}'$ can be described by optimality criteria using potential functions (see [17, Theorem 9.3]).

Proposition 9 *Suppose that in network $\mathcal{N}$, all excesses $b(v)$ and capacities u_{vw} are integers and cost functions c_{vw} are piecewise linear integral convex cost functions. Then there exists an integral* potential function $\pi : V \to \mathbb{Z}$ *such that a flow f is optimal if and only if it satisfies the following set of inequalities (so called* optimality conditions*):*

$$c_{vw}(f_{vw} + 1) - c_{vw}(f_{vw}) \ge \pi(w) - \pi(v) \ge c_{vw}(f_{vw}) - c_{vw}(f_{vw} - 1) \text{ for all } vw \in A.$$

Here $c_{vw}(-1)$ counts as $-\infty$ and $c_{vw}(u_{vw} + 1)$ counts as ∞, representing empty and saturated arcs.

The major drawback of the transformation is that it expands the network substantially. Note that the number of arcs in $\mathcal{N}'$ is typically not polynomial in $|V|$. Still, by a tricky scaling algorithm of [17, Section 14.5], one can determine the optimal integral flow and potential in (weakly) polynomial time.

Let U denote the largest arc capacity, C denote the largest cost function value and $S(|V|, |A|, C)$ denote the time needed for computing a shortest path in a network with $|V|$ nodes, $|A|$ arcs and cost functions bounded by C.

Theorem 10 ([17], Theorem 14.1) *There is a capacity scaling algorithm which obtains an integer optimal flow and an integer potential π for a convex cost flow problem in $O\left((|A| \cdot \log U) \cdot S(|V|, |A|, C)\right)$ time.*

Remark 11 *$S(|V|, |A|, C)$ is polynomial if $\log(C)$ is polynomial, see for example [19, Chapter 7].*

3 The halting problem on undirected multigraphs

Problem 12 (Chip-firing halting problem) *Given a graph G and $x \in Chip(G)$, determine whether x is terminating or non-terminating.*

In [16] Tardos proved that any terminating chip-firing game terminates within $O(|V|^2 M)$ moves. As a consequence, the trivial algorithm of playing a chip-firing game for at most $O(|V|^2 M)$ decides the halting problem. This bound is polynomial in the case of simple graphs but only pseudo-polynomial for multigraphs. We show an example of a chip-firing game which terminates, but not in polynomial time.

Example 13 *Let the multigraph G be described by the following adjacency matrix A_G. Consider the chip-firing game started from chip-distribution x.*

$$A_G = \begin{bmatrix} 0 & 2^n & 0 & 0 \\ 2^n & 0 & 1 & 0 \\ 0 & 1 & 0 & 4^n \\ 0 & 0 & 4^n & 0 \end{bmatrix} \qquad x = \begin{bmatrix} 2^{n+1} - 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

It is easy to check that the only possible firing sequence is obtained by the first and second node firing alternatingly. The game terminates after 2^n such turns.

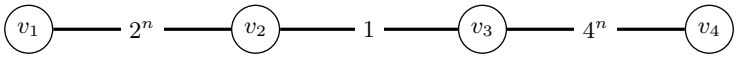


Figure 1: Example of a multigraph with a chip distribution which terminates, but not in polynomial time.

3.1 The halting problem is in NP∩co-NP

The following definition originates in the graph divisor theory of Baker and Norine [3].

Definition 14 *For $x, y \in \mathrm{Chip}(G)$, we say that x and y are* linearly equivalent *(and denote it by $x \sim y$) if there exists $z \in \mathbb{Z}^V$ such that $x = y + Lz$.*

Remark 15 *It is easy to check that $\sim$ is an equivalence relation on $\mathrm{Chip}(G)$. Note that x and y are linearly equivalent iff y is reachable from x in the unconstrained variant of chip-firing game, in which a node does not need to be active before it fires.*

The next lemma appeared in [6, Lemma 4.3.] and [10, Lemma 2.1].

Lemma 16 *Let G be a multigraph and $x, y \in \mathrm{Chip}(G)$. If $x \sim y$, then x is terminating if and only if y is terminating.*

To be self-contained, we include the proof from [14].

PROOF: By symmetry, it is enough to prove that if x is terminating, then y is also terminating.

Let x be a terminating chip-distribution. We play the chip-firing game starting from x until it terminates. Let the final configuration be x^*. Clearly, $x^* \sim x \sim y$. Let z be an integer vector with $x^* = y + Lz$. We can suppose that z is non-negative (otherwise we can add 1 to each of its coordinates, maintaining $x^* = y + Lz$). Now we start a *bounded* chip-firing game from y defined by the following rule: if there is a node v with at least $d(v)$ chips that has been fired less than $z(v)$ times, then one such node is fired. If there is no such node, the game ends. Clearly, after at most $\sum_{v \in V} z(v)$ firings, the bounded game terminates. We claim that the final distribution $y' = y + Lz'$ (where $z' \leq z$) is stable. Indeed, as the game stopped, for any node v with $y'(v) \geq d(v)$, $z'(v) = z(v)$. As x^* is stable, $x^*(v) < d(v)$, hence from $x^* = y' + L(z - z')$ and $z(v) = z'(v)$, we get $d(v) > x^*(v) \geq y'(v)$, which is a contradiction. □

Remark 17 *Linear equivalence is decidable in polynomial time, see [13] for details.*

Definition 18 *We call a chip-distribution $x \in \mathrm{Chip}(G)$* recurrent *if there exists a non-empty sequence of legal firings that transforms x to itself.*

The proof of [5, Theorem 3.3] implies the following proposition:

Proposition 19 *A chip-distribution $x \in \mathrm{Chip}(G)$ is recurrent if and only if there is an acyclic orientation $\mathcal{O}$ of G such that $\mathbf{d}^-_{\mathcal{O}} \leq x$.*

The following proposition was first formulated and proved in [13].

Proposition 20 *The chip-firing halting problem for undirected multigraphs is in* **NP∩co-NP**.

PROOF: **NP**: An efficiently verifiable proof for $x \in \mathrm{Chip}(G)$ being terminating is a stable chip-distribution y and an integer vector z such that $y = x + Lz$.

co-NP: An efficiently verifiable proof for $x \in \mathrm{Chip}(G)$ being non-terminating is a recurrent chip-distribution y' and an integer vector z' such that $y' = x + Lz'$ such that $y' = x + Lz'$. One can prove that y' is recurrent by giving an acyclic orientation $\mathcal{O}$ with in-degree vector $\mathbf{d}^-_{\mathcal{O}} \leq y'$. □

Remark 21 *In [13], a different argument is provided for showing that y' is recurrent (that one also works in the more general case of Eulerian directed multigraphs).*

3.2 A polynomial algorithm for the special case of M chips

In this section we present the main theorem of our paper, which is the following.

Theorem 22 *There is a polynomial algorithm deciding the halting problem for an undirected multigraph G and a distribution $x \in \mathrm{Chip}(G)$ with $\deg(x) = M$.*

The algorithm is based on Theorem 4.10. of [1], which is the following.

Theorem 23 (An–Baker–Kuperberg–Shokrieh) *Given any $x \in \mathrm{Chip}(G)$ with $\deg(x) = M$, there is an orientation $\mathcal{O}$ such that $x \sim \mathbf{d}^-_{\mathcal{O}}$.*
(The in-degree vector $\mathbf{d}^-_{\mathcal{O}}$ is considered as a chip-firing distribution here.)

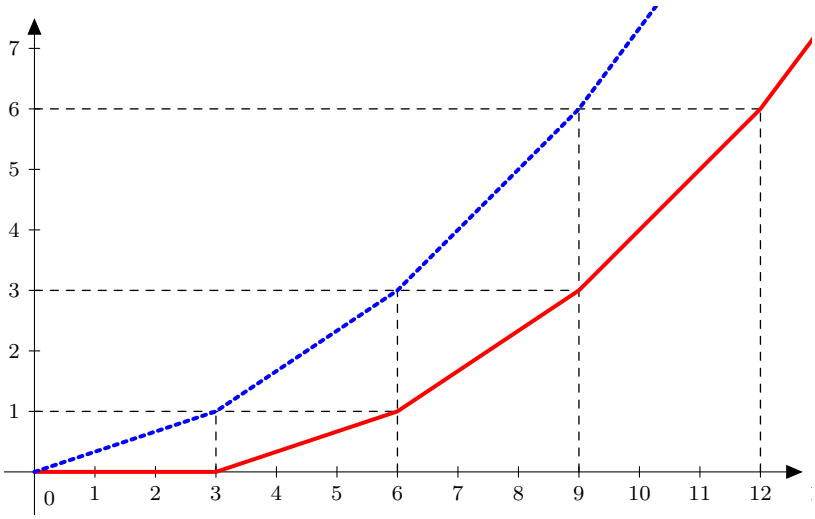


Figure 2: The cost function of the forward edge and the backward edge (dotted) obtained from an edge of G with multiplicity $m_{vw} = 3$.

Note that in [1], the theorem is stated for divisors of degree $g - 1 = M - n$, but they subtract 1 from the indegrees. Although multigraphs are not mentioned there, the original proof works for multigraphs without any difficulty. It is already noted in [1, Remark 4.14.] that while their proof is not algorithmic, Backman's *Unfurling algorithm* [2, Section 4] gives an algorithm which finds such an orientation. Notice that Backman's algorithm is polynomial for simple graphs but only pseudo-polynomial for multigraphs. Therefore we need a new algorithm to prove the following theorem:

Theorem 24 *Let G be a multigraph G and $x \in \mathrm{Chip}(G)$ with $\deg(x) = M$. Then there is a polynomial algorithm that computes an orientation $\mathcal{O}$ of G with $\mathbf{d}^-_{\mathcal{O}} \sim x$.*

Proof: Let $\mathcal{O}_1$ be an arbitrarily chosen consistent orientation of G. $\mathcal{O}_1$ being consistent, it defines a simple directed graph $G_1 = (V, A_1)$; a single arc $\overrightarrow{vw} \in A_1$ represents the multiedge vw if it is oriented from v to w in $\mathcal{O}_1$.

Let $d^-_1(v) = d^-_{\mathcal{O}_1}(v)$ denote the indegree of node v in the orientation $\mathcal{O}_1$. $\Gamma(v), \Gamma^+_1(v)$ and $\Gamma^-_1(v)$ denote the set of neighbors of v in undirected and directed sense:

$$\Gamma(v) = \{w : vw \in E\};\ \Gamma^+_1(v) = \{w : \overrightarrow{vw} \in A_1\};\ \Gamma^-_1(v) = \{w : \overrightarrow{wv} \in A_1\}.$$

Let $U = n \cdot \max_{v \in V}\{|x(v)|\}$. Now we are ready to define the auxiliary network $\mathcal{N}$.

1. The node set of $\mathcal{N}$ is V.
2. In $\mathcal{N}$ there is a *forward arc* $\overrightarrow{vw}$ (directed as in $\mathcal{O}_1$) and *backward arc* $\overrightarrow{wv}$ (directed oppositely as in $\mathcal{O}_1$) corresponding to every arc $\overrightarrow{vw} \in A_1$, both with capacity U.
3. The excess of node v is $b(v) = x(v) - d^-_1(v)$, for all $v \in V$.
4. We define a piecewise linear integral convex cost function on each arc $\overrightarrow{vw}$ of $\mathcal{N}$. On forward edges, the cost function is: $C^{\rightarrow}_{vw}(t) = \binom{\lfloor \frac{t}{m_{vw}} \rfloor}{2} + \left\lfloor \frac{t}{m_{vw}} \right\rfloor \left(\frac{t}{m_{vw}} - \left\lfloor \frac{t}{m_{vw}} \right\rfloor \right)$, which is the piecewise linear function with slope k between $k \cdot m_{vw}$ and $(k+1) \cdot m_{vw}$ for any $k \in \mathbb{N} \cup \{0\}$.

   On backward edges, the cost function is: $C^{\leftarrow}_{vw}(t) = \binom{\lfloor \frac{t}{m_{vw}} \rfloor + 1}{2} + \left(\left\lfloor \frac{t}{m_{vw}} \right\rfloor + 1 \right) \left(\frac{t}{m_{vw}} - \left\lfloor \frac{t}{m_{vw}} \right\rfloor \right)$, which is the piecewise linear function with slope k between $(k-1) \cdot m_{vw}$ and $k \cdot m_{vw}$ for any $k \in \mathbb{N}$.

As $C^{\rightarrow}_{vw}$ and $C^{\leftarrow}_{vw}$ are both piecewise linear integral convex cost functions, $\mathcal{N}$ satisfies the conditions of Proposition 9.

Hence we can use the algorithm of Theorem 10 to calculate a min cost flow f and potential $\pi : V \to \mathbb{Z}$. Note that the sum of all positive excesses is less than U, hence no arcs are saturated by a min cost flow.

Remark 25 *Note that the maximal capacity in the network is U and all capacities are bounded by $C' = \binom{U+1}{2}$. As $U \leq n \cdot M$, the bound given in Theorem 10 is polynomial in the input size. On the other hand, the $\log(U)$ factor means that we can only prove a weakly polynomial running time.*

For any arc $\overrightarrow{vw} \in A_1$, let $f^{\rightarrow}(vw)$ and $f^{\leftarrow}(vw)$ denote the value of f on the forward arc and backward arc assigned to $\overrightarrow{vw}$, respectively. Note that, as f is an optimal cost flow, at least one of $f^{\rightarrow}(vw)$ and $f^{\leftarrow}(vw)$ is 0.

Let $f(vw) = f^{\rightarrow}(vw) - f^{\leftarrow}(vw)$ and $f_\pi(vw) = f(vw) - [\pi(w) - \pi(v)]m_{vw}$. By the optimality conditions of Proposition 9,

$$[\pi(v) - \pi(u)]m_{vw} \leq f(vw) \leq ([\pi(w) - \pi(v)] + 1)\, m_{vw}$$

hence $0 \leq f_\pi(vw) \leq m_{vw}$.

Now we are ready to define the desired orientation $\mathcal{O}$: for any arc $\overrightarrow{vw} \in A_1$, we reverse $f_\pi(vw)$ pieces of $\overrightarrow{vw}$, i.e. in $\mathcal{O}$ the edge vw has $\overrightarrow{m}_{vw} = m_{vw} - f_\pi(vw)$ pieces oriented from v to w and $\overleftarrow{m}_{vw} = f_\pi(vw)$ pieces oriented from w to v. The following Claim 26 implies that $\mathbf{d}^-_{\mathcal{O}} \sim x$, i.e. $\mathcal{O}$ fulfills the requirements of Theorem 24. □

Claim 26 *The indegree vector of $\mathcal{O}$ is $\mathbf{d}^-_{\mathcal{O}}(v) = x - L\pi$.*

Proof: Consider any node $v \in V$. As f is a flow,

$$b(v) = x(v) - d^-_1(v) = \sum_{w \in \Gamma^+_1(v)} f(vw) - \sum_{w \in \Gamma^-_1(v)} f(wv). \tag{4}$$

By using the fact that $f(vw) = f_\pi(vw) + [\pi(w) - \pi(v)]\, m_{vw}$, we get

$$\begin{aligned}
&\sum_{w \in \Gamma^+_1(v)} f(vw) - \sum_{w \in \Gamma^-_1(v)} f(wv) = \\
= &\left(\sum_{w \in \Gamma^+_1(v)} f_\pi(vw) + \sum_{w \in \Gamma^+_1(v)} [\pi(w) - \pi(v)]\, m_{vw} \right) - \left(\sum_{w \in \Gamma^-_1(v)} f_\pi(wv) + \sum_{w \in \Gamma^-_1(v)} [\pi(v) - \pi(w)]\, m_{wv} \right) = \\
= &\sum_{w \in \Gamma^-_1(v)} f_\pi(vw) - \sum_{w \in \Gamma^-_1(v)} f_\pi(wv) + \sum_{w \in \Gamma(v)} [\pi(w) - \pi(v)]\, m_{vw}.
\end{aligned}$$

Reformulation and equation 4 gives:

$$d^-_1(v) + \sum_{w \in \Gamma^+_1(v)} f_\pi(vw) - \sum_{w \in \Gamma^-_1(v)} f_\pi(wv) = x(v) - \sum_{w \in \Gamma(v)} [\pi(w) - \pi(v)]\, m_{vw}.$$

Notice that by the definition of $\mathcal{O}$, the left hand side is exactly $\mathbf{d}^-_{\mathcal{O}}$. Hence

$$d^-_{\mathcal{O}}(v) = x(v) - \sum_{w \in \Gamma(v)} [\pi(v) - \pi(w)]\, m_{vw} = x(v) - L\pi(v)$$

for any v, which is exactly the statement of our claim. □

Remark 27 *The running time of the algorithm is dominated by the integral convex cost flow algorithm, which is polynomial by Remark 25.*

Remark 28 *Note that the idea of using flow algorithms has already appeared in Backman's paper [2, Algorithm 7.7]. The main difference in comparison to his work is that while he uses alternating phases of MFMC subroutines and cut reversing, we can compress these into only one flow cost minimizing subroutine. In Backman's algorithm, the number of phases is only pseudo-polynomial, so this improvement is essential in the case of large edge multiplicities.*

Another interesting feature of our algorithm is that the dual variable π has a meaning as a firing vector of an unconstrained chip-firing game.

Proof of Theorem 22: The algorithm consists of two main parts:

(P1) Compute an orientation $\mathcal{O}$ with $\mathbf{d}^-_{\mathcal{O}} \sim x$. This can be done in polynomial time by Theorem 24.

(P2) Decide whether the sink-reversal game started from $\mathcal{O}$ is terminating or non-terminating. This can be easily done using in $O(|E|)$ by Proposition 7.

By Lemma 16 and Claim 5, x is terminating if and only if the sink-reversal game started from $\mathcal{O}$ is terminating. □

3.2.1 Distributions with more than M chips

By [5, Theorem 3.3], the chip-firing halting problem is trivial if the number of chips is less than M. The special case solved in this paper is the minimal nontrivial case. Backman [2, Theorem 4.10] gave a generalization of Theorem 23. By reformulating it to the theory of chip-firing games, we get the following theorem.

Theorem 29 (Reformulation of Theorem 4.10. of [2]) *(1) If a chip-distribution x with $\deg(x) \geq M$ is terminating, then it is linearly equivalent to the in-degree vector of a sink-free partial orientation. (2) If a chip-distibution x with $\deg(x) \geq M$ is non-terminating, then it is linearly equivalent to a chip-distribution y such that $y \geq \mathbf{d}^-_{\mathcal{O}}$, where $\mathcal{O}$ is an acyclic partial orientation.*

Backman also gives an algorithm for computing these orientations. His algorithm is polynomial for simple graphs but only pseudo-polynomial for multigraphs. It can be the subject of some future work to upgrade this algorithm to run in polynomial time.

3.3 Connection to graph divisor theory

In [3], Baker and Norine established graph divisor theory (or Riemann–Roch-theory of graphs) as a discrete analogue of the classical Riemann–Roch-theory of algebraic curves. *Graph divisors* are integer-valued functions on the node set of a graph. Non-negative valued divisors are called *effective*. The basic question about a divisor is whether it is linearly equivalent to an effective divisor. This question is equivalent to the chip-firing halting problem, see [3, Section 5.5].

The standard way of deciding whether a divisor is equivalent to an effective divisor is by v_0-reducing it (for the definition of v_0-reduced divisors, see [3, 7]). In [7, Section 5.1], an algorithm is given for v_0-reducing a divisor of a multigraph. The algorithm is polynomial for simple graphs but only pseudo-polynomial for multigraphs.

Problem 30 *[11] Is there a polynomial algorithm for v_0-reducing divisors in multigraphs?*

An affirmative answer to Problem 30 would prove Conjecture 1.

Acknowledgements

The author would like to say thanks to András Frank for suggesting the idea of considering min cost flows; to Dion Gijswijt for calling the author's attention to the multigraphs case by Problem 30 and to Viktor Kiss for Example 13. Special thanks to Lilla Tóthmérész for fruitful discussions about the topic and many valuable comments about the manuscript.

References

[1] Y. An, M. Baker, G. Kuperberg and F. Shokrieh, Canonical representatives for divisor classes on tropical curves and the Matrix-Tree Theorem, *Forum of Mathematics, Sigma* **2** (2014)

[2] S. Backman, Riemann-Roch theory for graph orientations, *ArXiv Preprint*, `http://arxiv.org/abs/1401.3309` (2015)

[3] M. Baker and S. Norine, Riemann–Roch and Abel–Jacobi theory on a finite graph, *Adv. Math.* **215** p. 766–768 (2007)

[4] A. Björner and L. Lovász, Chip-firing games on directed graphs, *J. Algebraic Combin.* **1(4)** p. 305–328 (1992)

[5] A. Björner, L. Lovász and P. Shor, Chip-firing games on graphs, *European J. Combin.* **12(4)** p. 283–291 (1991)

[6] B. Bond and L. Levine, Abelian networks I. Foundations and examples, *SIAM J. Discrete Math.* **30(2)** p. 856–874 (2016)

[7] J. van Dobben de Bruyn, Reduced divisors and gonality in finite graphs, *Bachelor's thesis* Mathematisch Instituut, Universiteit Leiden, (2012)

[8] K. Eriksson, No Polynomial Bound for the Chip Firing Game on Directed Graphs, *Proc. Amer. Math. Soc.* **112** p. 1203–1205 (1991)

[9] D. Erdős, A. Frank and K. Kun, Sink-stable sets of digraphs, *SIAM J. Discrete Math.* **28(4)** p. 1651–1674 (2014)

[10] M. Farrell and L. Levine, CoEulerian graphs, *Proc. Amer. Math. Soc.* **144** p. 2847–2860 (2016)

[11] D. Gijswijt, Private communication (2013)

[12] E. Gioan, Enumerating degree sequences in digraphs and a cycle-cocycle reversing system, *European J. Combin.* **28(4)** p. 1351–1366 (2007)

[13] B. Hujter, V. Kiss and L. Tóthmérész, On the complexity of the chip-firing reachability problem, *Proc. Amer. Math. Soc.* electronically published on February 15, 2017, DOI: `https://doi.org/10.1090/proc/1349` (to appear in print).

[14] B. Hujter and L. Tóthmérész, Chip-firing based methods in the Riemann–Roch theory of directed graphs, *EGRES Technical Report* TR 16-01 (2016)

[15] V. Kiss and L. Tóthmérész, Chip-firing games on Eulerian digraphs and **NP**-hardness of computing the rank of a divisor on a graph, *Discrete Appl. Math.* **193** p. 48–56 (2015)

[16] G. Tardos, Polynomial Bound for a Chip Firing Game on Graphs, *SIAM J. Discrete Math.* **1(3)** p. 397–398 (1988)

[17] R. K. Ahuja, T. Magnanti and J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, *Prentice-Hall*, Inc. (1993)

[18] A. Frank, Connections in Combinatorial Optimization, *Oxford University Press* (2011)

[19] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency. Vol. A., Paths, flows, matchings. Chapters 1-38 *Springer-Verlag* (2003)

4 Appendix: The proof of the characterization of terminating sink-reversal games

The main goal of this appendix is to give a proof for Proposition 6:

Proposition 6 *A sink-reversal game started from orientation $\mathcal{O}$ of G is terminating if and only if $\mathcal{O}$ contains a directed cycle.*

The proof is a consequence of the forthcoming Claim 31 and Proposition 33. The proof is based on ideas from [9, Corollary 3.3.].

Claim 31 *A* sink-reversal game *started from an acyclic orientation $\mathcal{O}$ of G is non-terminating.*

Proof: When we reverse a sink node v in an acyclic orientation, we cannot get a new directed circle as all reversed edges are incident to v but v becomes a source in the new orientation. Hence the resulting orientation remains acyclic and therefore contains a sink. □

Next we turn our attention to orientations containing directed cycles.

Lemma 32 *If there is a directed path from v to a node w contained in a directed cycle then v can never be sink-reversed.*

Proof: As w is contained in a directed cycle, there is a directed walk:

$$v_0 = v \to v_1 \to \ldots \to v_{k-1} \to v_k = w \to v_{k+1} \to v_{\ell-1} \to v_\ell = w$$

None of the nodes of the sequence is originally a sink. v_{i+1} must be sink-reversed before the first sink-reversal of v_i, unless v_i cannot become a sink. But as $v_k = v_\ell$, it means that none of the nodes of the set $\{v_0, v_1, \ldots, v_\ell\}$ can be the first to be sink-reversed. □

Now let $\mathcal{O}$ be an orientation of the multigraph $G = (V, E)$. We define the weighted digraph $G^*_{\mathcal{O}}(V, A^*)$ the following way: if there is at least one edge oriented from v to w in $\mathcal{O}$ then we add an arc $\overrightarrow{vw}$ to A^* with weight 0 (forward arc). If there is at least one edge oriented from v to w in $\mathcal{O}$ then we add an arc $\overrightarrow{wv}$ to A^* with weight 1 (backward arc). Let $S = \{v \in V : v \text{ is contained in a directed cycle of } \mathcal{O}\}$.

Let us define the integer-valued potential function $\pi_{\mathcal{O}} : V \to Z$ the following way: let $\pi_{\mathcal{O}}(v)$ be the weight of the minimal weight dipath from S to v in $G^*_{\mathcal{O}}(V, A^*)$. In particular, $\pi_{\mathcal{O}}(v) = 0$ if and only if $v \in S$.

Proposition 33 *Let $\mathcal{O}$ be an orientation of G containing at least one directed cycle. The sink-reversal game started from orientation $\mathcal{O}$ is terminating and each node v is sink-reversed exactly $\pi_{\mathcal{O}}(v)$ times.*

Proof: If v is a sink node in $\mathcal{O}$, then $\pi_{\mathcal{O}}(v) > 0$ as no directed circle can be reached from v in $\mathcal{O}$. It can be easily checked that when the sink node v is reversed, $\pi_{\mathcal{O}}(v)$ decreases by 1 and for any other $w \in V$, $\pi_{\mathcal{O}}(w)$ remains unchanged.

The sink-reversal game cannot halt while there is a node with a positive $\pi_{\mathcal{O}}$ value. Indeed, if there is no sink node in $\mathcal{O}$, then from any node v, we can greedily find a dipath leading to a directed circle; so by Lemma 32; if there is any node v with $\pi_{\mathcal{O}}(v) > 0$, then there is a sink. □

Corollary 34 *Any terminating sink-reversal game terminates in less than $|V|^2$ sink-reversals.*

Proof: $\pi_{\mathcal{O}}(v) \leq |V| - 1$ for any $v \in V$. □

Remark 35 *Proposition 7 gives a way to decide whether a sink-reversal game is terminating or not. Another way is the following. Start a sink-reversal game from $\mathcal{O}$. By Corollary 34, if the game does not terminate in $|V|^2$ steps then it is non-terminating.*

The Quadratic M-Convexity Testing Problem [Extended Abstract]

YUNI IWAMASA*

Department of Mathematical Informatics,
Graduate School of Information Science and
Technology,
University of Tokyo
Tokyo, 113-8656, Japan.
yuni_iwamasa@mist.i.u-tokyo.ac.jp

Abstract: A function f on $\{0,1\}^n$ is said to be M-convex if for $x, y \in \{0,1\}^n$ and $i \in [n]$ with $x_i > y_i$, there exists $j \in [n]$ with $y_j > x_j$ satisfying $f(x)+f(y) \geq f(x-\chi_i+\chi_j)+f(y+\chi_i-\chi_j)$, where χ_i is the ith unit vector. M-convex functions play a central role in discrete convex analysis.

In this paper, we consider the quadratic M-convexity testing problem (QMCTP). This is the problem of deciding whether a given form $\sum_{i\in[n]} a_i x_i + \sum_{1\leq i<j\leq n} a_{ij} x_i x_j$ on $\sum_{i\in[n]} x_i = r$ is M-convex, where $a_i \in \mathbf{R}$ and $a_{ij} \in \mathbf{R}\cup\{+\infty\}$. It is known that if every a_{ij} takes a finite value, then (QMCTP) can be solved in polynomial time. We show that (QMCTP) is co-NP complete in general, and but is polynomial-time solvable under a mild assumption. Furthermore, we propose an $O(n^2)$-time algorithm for solving (QMCTP) under the assumption.

Keywords: discrete convex analysis, M-convex, testing problem

1 Introduction

A function f on $\{0,1\}^n$ is said to be *M-convex* [6] if it satisfies the following exchange axiom:

Exchange Axiom: For $x, y \in \{0,1\}^n$ and $i \in \mathrm{supp}(x) \setminus \mathrm{supp}(y)$, there exists $j \in \mathrm{supp}(y) \setminus \mathrm{supp}(x)$ such that

$$f(x) + f(y) \geq f(x - \chi_i + \chi_j) + f(y + \chi_i - \chi_j),$$

where $\mathrm{supp}(x) := \{i \mid x_i = 1\}$ for $x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ and χ_i is the ith unit vector. In general, M-convex functions are defined on the integer lattice $\mathbf{Z}^n$. In this paper, we restrict ourselves to consider M-convex functions defined on $\{0,1\}^n$, which are equivalent to negative of *valuated matroids* introduced by Dress–Wenzel [2, 3]. M-convex functions play a central role in *discrete convex analysis* [7]. Indeed, M-convex functions appear in many areas such as operations research, economics, and game theory (see e.g., [7, 8, 9]). Quadratic M-convex functions also appear in many areas, and constitute a basic and important class of discrete functions. Quadratic M-convex functions have a close relationship with *tree metrics* [4], which is an important concept for mathematical analysis in phylogenetics (see e.g., [11]). Recently, Iwamasa–Murota–Živný [5] have revealed hidden quadratic M-convexity in valued constraint satisfaction problems with joint winner property, and presented a perspective to their polynomial-time solvability from discrete convex analysis.

In this paper, we consider the *quadratic M-convexity testing problem (QMCTP)* defined as follows. Let $\overline{\mathbf{R}} := \mathbf{R} \cup \{+\infty\}$ and $[n] := \{1, 2, \ldots, n\}$ for a positive integer n with $n \geq 4$.

*This research was supported by JSPS Research Fellowship for Young Scientists.

Given: $a_i \in \mathbf{R}$ for $i \in [n]$, $a_{ij} \in \overline{\mathbf{R}}$ for $1 \leq i < j \leq n$, and a positive integer r with $2 \leq r \leq n-2$.
Question: Is the quadratic function $f : \{0,1\}^n \to \overline{\mathbf{R}}$ defined by

$$f(x_1, x_2, \ldots, x_n) := \begin{cases} \sum_{i \in [n]} a_i x_i + \sum_{1 \leq i < j \leq n} a_{ij} x_i x_j & \text{if } \sum_{i \in [n]} x_i = r, \\ +\infty & \text{otherwise} \end{cases} \tag{1}$$

M-convex?

Here we assume that $a_{ij} = a_{ji}$ for distinct $i, j \in [n]$ and the effective domain $\operatorname{dom} f := \{x \in \{0,1\}^n \mid f(x) \text{ takes a finite value}\}$ is nonempty. In this paper, functions can take the infinite value $+\infty$, where $a < +\infty$, $a + \infty = +\infty$ for $a \in \mathbf{R}$, and $0 \cdot (+\infty) = 0$. In the case where a_{ij} takes a finite value for all distinct $i, j \in [n]$, the following theorem is immediate from [7, Theorem 6.4] (see also [7, Proposition 6.8]).

Theorem 1 ([7]; see also [10, Theorem 5.2]) *Suppose that a_{ij} takes a finite value for all distinct $i, j \in [n]$. Then a function of the form* (1) *is M-convex if and only if*

$$a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\} \tag{2}$$

holds for every distinct $i, j, k, l \in [n]$.

By Theorem 1, if a_{ij} is a finite value for all distinct $i, j \in [n]$, then (QMCTP) is solvable in polynomial time. However, if a_{ij} can take the infinite value $+\infty$ for some distinct $i, j \in [n]$, there exists an example such that the condition (2) does not characterize M-convexity. Indeed, define $f : \{0,1\}^5 \to \overline{\mathbf{R}}$ by

$$f(x_1, x_2, x_3, x_4, x_5) := \begin{cases} x_1x_3 + 2x_1x_4 + (+\infty) \cdot x_1x_5 + x_3x_5 + 2x_4x_5 & \text{if } \sum_i x_i = 3, \\ +\infty & \text{otherwise.} \end{cases} \tag{3}$$

Then f is M-convex; this can be verified by the definition of M-convexity. However, the condition (2) is violated since $a_{12} + a_{34} < \min\{a_{13} + a_{24}, a_{14} + a_{23}\}$ with $a_{12} + a_{34} = 0$, $a_{13} + a_{24} = 1$, and $a_{14} + a_{23} = 2$. Thus, in the general case, the complexity of (QMCTP) is not settled yet.

In this paper, we settle this problem by showing the following negative result.

Theorem 2 (QMCTP) *is co-NP complete.*

We also prove a positive result under the following seemingly natural condition.

Condition A: For any $i \in [n]$, there exists $x \in \operatorname{dom} f$ with $x_i = 1$.

Theorem 3 *If* Condition A *holds,* (QMCTP) *is solvable in $O(n^2)$ time.*

The rest of this paper is organized as follows. In Section 2, we prove Theorem 2. By the necessity of M-convexity under Condition A, we can classify functions into three types. In Section 3, we present a characterization of M-convexity under Condition A for three types. This characterization implies Theorem 1. In Section 4, we propose $O(n^2)$-time algorithms for (QMCTP) for each type. The proofs of the validity of these algorithms will be provided in the full version of this paper.

2 Co-NP Completeness of (QMCTP)

In this section, we show the co-NP completeness of (QMCTP) in the general case. In order to show Theorem 2, we prepare some lemmas.

In the terminology of discrete convex analysis, $X \subseteq \{0,1\}^n$ is said to be *M-convex* if for $x, y \in X$ and $i \in \operatorname{supp}(x) \setminus \operatorname{supp}(y)$, there exists $j \in \operatorname{supp}(y) \setminus \operatorname{supp}(x)$ such that $x - \chi_i + \chi_j, y + \chi_i - \chi_j \in X$. That is, an M-convex set X is nothing but the base family of some matroid. Note that if f is M-convex, then $\operatorname{dom} f$ is M-convex.

Lemma 4 *Suppose that f is a function of the form* (1) *such that* dom f *is M-convex. For some distinct $i, j \in [n]$, assume that there exist $x, y \in$ dom f with $x_i = 1$ and $y_j = 1$. Then, if $a_{ij} < +\infty$, there exists $z \in$ dom f with $z_i = z_j = 1$.*

PROOF: Take $x, y \in$ dom f with $|\mathrm{supp}(x) \setminus \mathrm{supp}(y)|$ minimum satisfying $x_i = y_j = 1$. It suffices to show $|\mathrm{supp}(x) \setminus \mathrm{supp}(y)| = 0$. Suppose, to the contrary, that $|\mathrm{supp}(x) \setminus \mathrm{supp}(y)| > 0$. First we assume $|\mathrm{supp}(x) \setminus \mathrm{supp}(y)| \geq 2$. Then there exists $i' \neq i$ such that $i' \in \mathrm{supp}(x) \setminus \mathrm{supp}(y)$. By the M-convexity of dom f for x, y, and i', there exists $j' \in \mathrm{supp}(y) \setminus \mathrm{supp}(x)$ such that $x - \chi_{i'} + \chi_{j'} \in$ dom f. If $j' = j$, then $x' := x - \chi_{i'} + \chi_j$ satisfies $x'_i = x'_j = 1$, a contradiction. If $j' \neq j$, then $x' := x - \chi_{i'} + \chi_{j'}$ satisfies $x'_i = y_j = 1$ and $|\mathrm{supp}(x') \setminus \mathrm{supp}(y)| < |\mathrm{supp}(x) \setminus \mathrm{supp}(y)|$. This is also a contradiction to the minimality of x and y. Hence we have $|\mathrm{supp}(x) \setminus \mathrm{supp}(y)| = 1 = |\mathrm{supp}(y) \setminus \mathrm{supp}(x)|$.

Since $x, y \in$ dom f, it holds that $a_{kl}, a_{ik}, a_{jk} < +\infty$ for any $k, l \in \mathrm{supp}(x - \chi_i) (= \mathrm{supp}(y - \chi_j))$. Moreover, we have $a_{ij} < +\infty$ by the assumption. Hence we obtain $z := x - \chi_k + \chi_j \in$ dom f for $k \in \mathrm{supp}(x - \chi_i)$. Hence z satisfies $z_i = z_j = 1$, a contradiction. Thus, we have $|\mathrm{supp}(x) \setminus \mathrm{supp}(y)| = 0$. □

For a function f of the form (1), we define an undirected graph $G_f = ([n], E_f)$ by $E_f := \{\{i, j\} \mid i, j \in [n],\ i \neq j,\ a_{ij} = +\infty\}$.

Lemma 5 *Suppose that* Condition A *holds. Then* dom f *is an M-convex set if and only if each connected component of G_f is a complete graph.*

PROOF: (if part). Let $A_1, A_2, \ldots, A_m$ be the connected components of G_f. Then dom f is represented by dom $f = \{x \in \{0, 1\}^n \mid \sum_i x_i = r,\ |x \cap A_p| \leq 1 \text{ for all } p \in [m]\}$. Hence dom f can be considered as the base family of a partition matroid. This implies that dom f is M-convex.

(only-if part). We prove the contrapositive. Suppose that some connected component of G_f is not complete. That is, there exist distinct $i, j, k \in [n]$ such that $\{i, j\}, \{j, k\} \in E_f$ and $\{i, k\} \notin E_f$. By Condition A, $a_{ik} < +\infty$, and Lemma 4, there exists $x \in$ dom f with $x_i = x_k = 1$.

Take any $x, y \in$ dom f with $x_i = x_k = 1$ and $y_j = 1$. Since $a_{ij} = a_{jk} = +\infty$, we have $\mathrm{supp}(x) \setminus \mathrm{supp}(y) \supseteq \{i, k\}$. Then for all $j' \in \mathrm{supp}(y) \setminus \mathrm{supp}(x)$, it holds that $x - \chi_i + \chi_{j'} \notin$ dom f or $y + \chi_i - \chi_{j'} \notin$ dom f. Indeed, if $j' = j$, then $x - \chi_i + \chi_j \notin$ dom f holds from $a_{kj} = +\infty$, and if $j' \neq j$, then $y + \chi_i - \chi_{j'} \notin$ dom f holds from $a_{ij} = +\infty$. This implies that dom f is not M-convex. □

Here we consider the following problem (P):

Given: A graph $G = (V, E)$ having an independent set with cardinality r.

Question: Let $T := \bigcup\{S \subseteq V \mid S \text{ is an independent set of } G \text{ with } |S| = r\}$. Is each connected component of the subgraph of G induced by T a complete graph?

Lemma 6 *The problem* (P) *is co-NP complete.*

PROOF: It is clear that the problem (P) is in co-NP. We show the co-NP hardness of (P) by reduction from the independent set problem, which is an NP-complete problem: Given $G = (V, E)$ and a positive integer $k \leq |V|$, we determine whether G contains an independent set of size at least k. For a given graph $G = (V, E)$ and a positive integer m, define $G_m := (V \cup V_m, E \cup E_m)$ by $|V_m| = m$, $V_m \cap V = \emptyset$, and $E_m := \{\{i, j\} \mid i \in V,\ j \in V_m\}$. Let $T_m := \bigcup\{S \subseteq V \mid S \text{ is an independent set of } G_m \text{ with } |S| = m\}$. Since $T_m \supseteq V_m$, each connected component of the subgraph of G_m induced by T_m is complete if and only if G does not have an independent set with cardinality at least m. Therefore we have the cardinality of a maximum independent set of G by solving (P) for G_k ($k = |V|, |V| - 1, \ldots, 1$). Indeed, the first k such that we output "no" by solving (P) for G_k is equal to the cardinality of a maximum independent set. Since the maximum independent set problem has a polynomial-time reduction to (P), (P) is co-NP hard. □

We are now ready to prove Theorem 2.

PROOF:[Proof of Theorem 2] It is clear that (QMCTP) is in co-NP. We show the co-NP hardness of (QMCTP) by reduction from the problem (P). Let $G = ([n], E)$ be a graph having an independent set with cardinality r. We define f_G by

$$f_G(x_1, x_2, \ldots, x_n) := \begin{cases} \sum_{1 \le i < j \le n} a_{ij} x_i x_j & \text{if } \sum_{i \in [n]} x_i = r, \\ +\infty & \text{otherwise,} \end{cases}$$

where $a_{ij} := +\infty$ for $\{i, j\} \in E$ and $a_{ij} := 0$ for $\{i, j\} \notin E$. Note that $x \in \mathrm{dom}\, f$ if and only if $\mathrm{supp}(x)$ is an independent set of G. We have $\mathrm{dom}\, f_G \neq \emptyset$ by the assumption that G has an independent set with cardinality r. We define X by $X := \bigcup \{x \in \{0,1\}^n \mid \mathrm{supp}(x)$ is an independent set of G with $|\mathrm{supp}(x)| = r\}$. Then there exists $x \in \mathrm{dom}\, f_G$ with $x_i = 1$ if and only if $i \in X$. For $x \in \{0,1\}^X$, define $\tilde{x} \in \{0,1\}^n$ by $\tilde{x}_i := x_i$ if $i \in X$ and $\tilde{x}_i := 0$ if $i \in [n] \setminus X$. Moreover define $f_G|_X(x) := f_G(\tilde{x})$ for $x \in \{0,1\}^X$. By the definition of X, f_G is M-convex (i.e., $\mathrm{dom}\, f_G$ is M-convex) if and only if $f_G|_X$ is M-convex (i.e., $\mathrm{dom}\, f_G$ is M-convex). Furthermore, by Lemma 5, $f_G|_X$ is M-convex if and only if each connected component of the subgraph of G induced by X is complete. This means that we can solve (P) by solving (QMCTP) for f_G. □

3 Characterization of Quadratic M-Convexity

In this section, we present a characterization of M-convexity under Condition A, which implies Theorem 1. By Lemma 5, we see that the following Condition B is necessary for the M-convexity.

Condition B: Each connected component of G_f is a complete graph.

Therefore, in this section, we can assume that a function f of the form (1) satisfies Conditions A and B. Let $A_1, A_2, \ldots, A_m$ be the vertex sets of the connected components of G_f with at least one edge, and define $A_0 := [n] \setminus \bigcup_{p=1}^{m} A_p$, which denotes the set of isolated vertices. Then we classify the types of f as follows.

Type I: $|A_0| + m \ge r + 2$.

Type II: $|A_0| + m = r + 1$.

Type III: $|A_0| + m = r$.

If $|A_0| + m < r$, then we have $\mathrm{dom}\, f = \emptyset$. Hence we exclude this case.

Theorem 7 (I): *A function f of Type* I *is M-convex if and only if it holds that*

$$a_{ij} + a_{kl} \ge \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\},$$

for every distinct $i, j, k, l \in [n]$.

(II): *A function f of Type* II *is M-convex if and only if it holds that*

$$a_{ij} + a_{kl} = a_{il} + a_{jk},$$

for every $p \in [m]$, distinct $i, k \in A_p$, and distinct $j, l \in [n] \setminus A_p$.

(III): *A function f of Type* III *is M-convex if and only if it holds that*

$$a_{ij} + a_{kl} = a_{il} + a_{jk},$$

for every distinct $p, q \in [m]$, distinct $i, k \in A_p$, and distinct $j, l \in A_q$.
Moreover, if f is an M-convex function of Type II *or* III*, then f is a linear function on* dom f.

We note that the function defined in (3) is of Type II. If a_{ij} is finite value for all distinct $i, j \in [n]$, the function f is of Type I. Hence Theorem 7 implies Theorem 1 as the finite case. By Theorem 7, we see that (QMCTP) is solvable in polynomial time under Condition A.

In the proof of Theorem 7, we use the following facts about the local exchange axiom characterizing M-convexity, which are immediate corollaries of [7, Theorem 6.4] (see also [7, Proposition 6.8]).

Theorem 8 ([7]) *A function $f : \{0,1\}^n \to \overline{\mathbf{R}}$ with* dom $f \subseteq \{x \in \{0,1\}^n \mid \sum_i x_i = r\}$ *is M-convex if and only if*

$$\begin{aligned} &f(z+\chi_i+\chi_j)+f(z+\chi_k+\chi_l) \\ &\quad \geq \min\{f(z+\chi_i+\chi_k)+f(z+\chi_j+\chi_l), f(z+\chi_i+\chi_l)+f(z+\chi_j+\chi_k)\} \end{aligned}$$

holds for all $z \in \{0,1\}^n$ and all distinct $i, j, k, l \in [n]$ such that $z+\chi_i+\chi_j, z+\chi_k+\chi_l \in$ dom f.

Lemma 9 *A function f of the form (1) is M-convex if and only if for every distinct $i, j, k, l \in [n]$ such that there exists $z \in \{0,1\}^n$ with $z+\chi_i+\chi_j, z+\chi_k+\chi_l \in$* dom f, *it holds that*

$$a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\}.$$

Note that, by Lemma 9, the condition (2) in Theorem 1 is sufficient for M-convexity. However, this is not necessary in general.

Proof:[Proof of Lemma 9] Take any $z \in \{0,1\}^n$ and distinct $i, j, k, l \in [n]$ such that $z+\chi_i+\chi_j, z+\chi_k+\chi_l \in$ dom f. By Theorem 8, it suffices to show that for such i, j, k, l,

$$\begin{aligned} &f(z+\chi_i+\chi_j)+f(z+\chi_k+\chi_l) \\ &\quad \geq \min\{f(z+\chi_i+\chi_k)+f(z+\chi_j+\chi_l), f(z+\chi_i+\chi_l)+f(z+\chi_j+\chi_k)\} \end{aligned}$$

holds if and only if $a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\}$ holds (note that the inequality $a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\}$ is independent of the choice of z).

Define $g : \{0,1\}^n \to \overline{\mathbf{R}}$ by

$$g(x) := \sum_{i \in [n]} a_i x_i + \sum_{1 \leq i < j \leq n} a_{ij} x_i x_j$$

for $x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$. Then we have

$$f(z+\chi_i+\chi_j) = g(z) + a_i + a_j + \sum_{p \in \mathrm{supp}(z)} a_{ip} + \sum_{p \in \mathrm{supp}(z)} a_{jp} + a_{ij}, \tag{4}$$

$$f(z+\chi_k+\chi_l) = g(z) + a_k + a_l + \sum_{p \in \mathrm{supp}(z)} a_{kp} + \sum_{p \in \mathrm{supp}(z)} a_{lp} + a_{kl}. \tag{5}$$

Since $f(z+\chi_i+\chi_j)$ and $f(z+\chi_k+\chi_l)$ take finite values, each term of (4) and (5), i.e., $g(z)$, a_{ij}, a_{kl}, and $a_{ip}, a_{jp}, a_{kp}, a_{lp}$ for $p \in \mathrm{supp}(z)$, also takes a finite value. Hence we obtain

$$\begin{aligned} &f(z+\chi_i+\chi_j)+f(z+\chi_k+\chi_l) \\ &\quad \geq \min\{f(z+\chi_i+\chi_k)+f(z+\chi_j+\chi_l), f(z+\chi_i+\chi_l)+f(z+\chi_j+\chi_k)\} \\ \Leftrightarrow\ &a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\}. \end{aligned}$$

□

A function f is said to be *M-concave* if $-f$ is M-convex. The following theorem (M-separation theorem) holds.

Theorem 10 ([7, Theorem 8.15]) *Suppose that f is M-convex and g is M-concave satisfying* dom $f \cap$ dom $g \neq \emptyset$ *and* $g(x) \leq f(x)$ *for any* $x \in$ dom $f \cap$ dom g. *Then there exist* $\alpha^* \in \mathbf{R}$ *and* $p^* \in \mathbf{R}^n$ *such that*

$$g(x) \leq \alpha^* + \sum_{i \in [n]} p_i^* x_i \leq f(x) \qquad (x \in \text{dom } f \cap \text{dom } g).$$

We are now ready to prove Theorem 7.

Proof:[Proof of Theorem 7] First we show the characterization of M-convexity. For $i \in [n]$ denote by B_i the connected component of G_f containing i. That is, $B_i = \{i\}$ for $i \in A_0$, and $B_i = A_p$ for $i \in A_p$. Note that $x \in$ dom f if and only if $\sum_i x_i = r$ and $|\text{supp}(x) \cap A_p| \leq 1$ for $p \in [m]$, and that if $a_{ij} = +\infty$ or $a_{kl} = +\infty$, it holds that $f(z + \chi_i + \chi_j) = +\infty$ or $f(z + \chi_k + \chi_l) = +\infty$ for all $z \in \{0,1\}^n$. In the following, we consider each type in turn.

Type I. We see that for all distinct $i, j, k, l \in [n]$ with $a_{ij} < +\infty$ and $a_{kl} < +\infty$, there exists $z \in \{0,1\}^n$ such that $z+\chi_i+\chi_j, z+\chi_k+\chi_l \in$ dom f. $|A_0 \setminus (B_i \cup B_j \cup B_k \cup B_l)| + |\{A_1, A_2, \ldots, A_m\} \setminus \{B_i, B_j, B_k, B_l\}| \geq r - 2$ holds since $|A_0| + m \geq r + 2$. Therefore we can take $z \in \{0,1\}^n$ satisfying $\text{supp}(z) \subseteq [n] \setminus (B_i \cup B_j \cup B_k \cup B_l)$, $|\text{supp}(z) \cap A_p| \leq 1$ for $p \in [m]$, and $\sum_i z_i = r - 2$. Then $z + \chi_i + \chi_j, z + \chi_k + \chi_l \in$ dom f holds for such z.

By Lemma 9, f is M-convex if and only if for every distinct $i, j, k, l \in [n]$ with $a_{ij}, a_{kl} < +\infty$, it holds that $a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\}$. Moreover, if $a_{ij} = +\infty$ or $a_{kl} = +\infty$, then $a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\}$ automatically holds. Hence f is M-convex if and only if for every distinct $i, j, k, l \in [n]$, it holds that $a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\}$.

Type II. We see that for distinct $i, j, k, l \in [n]$ with $a_{ij}, a_{kl} < +\infty$, there exists $z \in \{0,1\}^n$ such that $z + \chi_i + \chi_j, z + \chi_k + \chi_l \in$ dom f if and only if $(B_i \cup B_j) \cap (B_k \cup B_l) \neq \emptyset$ holds (note that we have $B_i \cap B_j = B_k \cap B_l = \emptyset$ since $a_{ij}, a_{kl} < +\infty$).

Suppose $(B_i \cup B_j) \cap (B_k \cup B_l) = \emptyset$ (i.e., B_i, B_j, B_k, and B_l are all disjoint). Then $|A_0| + m \geq 4$. Hence $r \geq 3$ since $|A_0| + m = r + 1$. Furthermore we obtain $|A_0 \setminus (B_i \cup B_j \cup B_k \cup B_l)| + |\{A_1, A_2, \ldots, A_m\} \setminus \{B_i, B_j, B_k, B_l\}| = r - 3$. Hence for all $z \in \{0,1\}^n$ such that $\sum_i z_i = r - 2$ and $|\text{supp}(z) \cap A_p| \leq 1$ for $p \in [m]$ with $A_p \neq B_i, B_j, B_k, B_l$, it holds that $|\text{supp}(z) \cap (B_i \cup B_j \cup B_k \cup B_l)| \neq \emptyset$. This means $z + \chi_i + \chi_j \notin$ dom f or $z + \chi_k + \chi_l \notin$ dom f. Thus, for $i, j, k, l \in [n]$ with $(B_i \cup B_j) \cap (B_k \cup B_l) = \emptyset$, there is no z satisfying $z + \chi_i + \chi_j, z + \chi_k + \chi_l \in$ dom f.

Suppose $(B_i \cup B_j) \cap (B_k \cup B_l) \neq \emptyset$. Without loss of generality, we also suppose $B_i \cap B_k \neq \emptyset$. Then there exists $p \in [m]$ such that $B_i = B_k = A_p$. Since $|A_0| + m = r + 1$, we have $|A_0 \setminus (B_i \cup B_j \cup B_k \cup B_l)| + |\{A_1, A_2, \ldots, A_m\} \setminus \{B_i, B_j, B_k, B_l\}| = |A_0 \setminus (B_j \cup B_l)| + |\{A_1, A_2, \ldots, A_m\} \setminus \{A_p, B_j, B_l\}| \geq r - 2$. Therefore we can take $z \in \{0,1\}^n$ satisfying $\text{supp}(z) \subseteq [n] \setminus (B_i \cup B_j \cup B_k \cup B_l)$, $|\text{supp}(z) \cap A_p| \leq 1$ for $p \in [m]$, and $\sum_i z_i = r - 2$. Then $z + \chi_i + \chi_j, z + \chi_k + \chi_l \in$ dom f holds for such z.

By Lemma 9, f is M-convex if and only if for every $p \in [m]$, distinct $i, k \in A_p$, and distinct $j, l \in [n] \setminus A_p$, it holds that $a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\}$. Since $a_{ik} = +\infty$, the above inequality can be represented as $a_{ij} + a_{kl} \geq a_{il} + a_{jk}$. Moreover, by replacing j with l, we have $a_{ij} + a_{kl} \leq a_{il} + a_{jk}$. Hence f is M-convex if and only if for every $p \in [m]$, distinct $i, k \in A_p$, and distinct $j, l \in [n] \setminus A_p$, it holds that $a_{ij} + a_{kl} = a_{il} + a_{jk}$.

Type III. We see that for distinct $i, j, k, l \in [n]$ with $a_{ij}, a_{kl} < +\infty$, there exists $z \in \{0,1\}^n$ such that $z + \chi_i + \chi_j, z + \chi_k + \chi_l \in$ dom f if and only if $B_i \cup B_j = B_k \cup B_l$ holds.

Suppose $B_i \cup B_j \neq B_k \cup B_l$. Without loss of generality, we also suppose $B_i \neq B_k$ and $B_i \neq B_l$. Then $|A_0| + m \geq 3$. Hence $r \geq 3$ since $|A_0| + m = r$. Furthermore we obtain $|A_0 \setminus (B_i \cup B_j \cup B_k \cup B_l)| + |\{A_1, A_2, \ldots, A_m\} \setminus \{B_i, B_j, B_k, B_l\}| \leq |A_0 \setminus (B_i \cup B_k \cup B_l)| + |\{A_1, A_2, \ldots, A_m\} \setminus \{B_i, B_k, B_l\}| = r - 3$. Hence for all $z \in \{0,1\}^n$ such that $\sum_i z_i = r - 2$ and $|\text{supp}(z) \cap A_p| \leq 1$ for $p \in [m]$ with $A_p \neq B_i, B_j, B_k, B_l$, it holds that $|\text{supp}(z) \cap (B_i \cup B_j \cup B_k \cup B_l)| \neq \emptyset$. This means $z + \chi_i + \chi_j \notin$ dom f or $z + \chi_k + \chi_l \notin$ dom f. Thus, for $i, j, k, l \in [n]$ with $(B_i \cup B_j) \cap (B_k \cup B_l) = \emptyset$, there is no z satisfying $z + \chi_i + \chi_j, z + \chi_k + \chi_l \in$ dom f.

Suppose $B_i \cup B_j = B_k \cup B_l$. Without loss of generality, we also suppose $B_i = B_k$ and $B_j = B_l$. Then there exist distinct $p, q \in [m]$ such that $B_i = B_k = A_p$ and $B_j = B_l = A_q$. Since $|A_0| + m = r$, we have $|A_0 \setminus (B_i \cup B_j \cup B_k \cup B_l)| + |\{A_1, A_2, \ldots, A_m\} \setminus \{B_i, B_j, B_k, B_l\}| = |A_0| + |\{A_1, A_2, \ldots, A_m\} \setminus \{A_p, A_q\}| = r - 2$. Therefore we can take $z \in \{0, 1\}^n$ satisfying $\mathrm{supp}(z) \subseteq [n] \setminus (B_i \cup B_j \cup B_k \cup B_l)$, $|\mathrm{supp}(z) \cap A_p| \leq 1$ for $p \in [m]$, and $\sum_i z_i = r - 2$. Then $z + \chi_i + \chi_j, z + \chi_k + \chi_l \in \mathrm{dom}\ f$ holds for such z.

By Lemma 9, f is M-convex if and only if for every distinct $p, q \in [m]$, distinct $i, k \in A_p$, and distinct $j, l \in A_q$, it holds that $a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\}$. Since $a_{ik} = a_{jl} = +\infty$, the above inequality can be represented as $a_{ij} + a_{kl} \geq a_{il} + a_{jk}$. Moreover, by replacing j with l, we have $a_{ij} + a_{kl} \leq a_{il} + a_{jk}$. Hence f is M-convex if and only if for every distinct $p, q \in [m]$, distinct $i, k \in A_p$, and distinct $j, l \in A_q$, it holds that $a_{ij} + a_{kl} = a_{il} + a_{jk}$.

Linearity. Then we show linearity of an M-convex function f of Type II or III. By the characterization of Type II or III, the function g defined by

$$g(x) := \begin{cases} f(x) & \text{if } f(x) < +\infty, \\ -\infty & \text{if } f(x) = +\infty \end{cases}$$

is M-concave for an M-convex function f of Type II or III. By Theorem 10, there exist $\alpha^* \in \mathbf{R}$ and $p^* \in \mathbf{R}^n$ such that

$$f(x) = g(x) \leq \alpha^* + \sum_{i \in [n]} p_i^* x_i \leq f(x) \qquad (x \in \mathrm{dom}\ f).$$

This means that f is a linear function on dom f. □

4 Testing Quadratic M-Convexity in Quadratic Time

In this section, we present an $O(n^2)$-time algorithm for (QMCTP) under the assumption that a function f of the form (1) satisfies Condition A (and Condition B). As seen in Section 3, f is classified into Types I, II, or III. For Types I, II, and III, we give Algorithms I, II, and III, respectively.

By Theorem 7, we see that the M-convexity of a function of the form (1) depends only on quadratic coefficients $(a_{ij})_{i,j \in [n]}$. We say that a function f of the form (1) is defined by $(a_{ij})_{i,j \in [n]}$ if the quadratic coefficients of f is equal to $(a_{ij})_{i,j \in [n]}$. We also say that $(a_{ij})_{i,j \in [n]}$ satisfies the *anti-tree metric property* if $a_{ij} + a_{kl} \geq \min\{a_{ik} + a_{jl}, a_{il} + a_{jk}\}$ holds for all distinct $i, j, k, l \in [n]$, and that $(a_{ij})_{i,j \in [n]}$ satisfies the *anti-ultrametric property* if $a_{ij} \geq \min\{a_{ik}, a_{jk}\}$ holds for all distinct $i, j, k \in [n]$. Note that the anti-tree metric property characterizes the M-convexity of functions of Type I (cf. Theorem 7).

Algorithm I (for Type I).

Step 1: Define $\alpha := \min\{a_{ij} \mid i, j \in [n]\}$, $b_i := \min\{a_{ij} \mid j \in [n] \setminus \{i\}\} - \alpha$ for $i \in [n]$, and $\hat{a}_{ij} := a_{ij}$ for distinct $i, j \in [n]$.

Step 2: Update $\hat{a}_{ij} \leftarrow \hat{a}_{ij} - b_i - b_j$ for distinct $i, j \in [n]$.

Step 3: If $(\hat{a}_{ij})_{i,j \in [n]}$ satisfies the anti-ultrametric property, output that "f is M-convex." Otherwise, output that "f is not M-convex." □

Algorithm II (for Type II).

Step 1: Define $\alpha := \min\{a_{ij} \mid i, j \in [n]\}$ and $\hat{a}_{ij} := a_{ij}$ for distinct $i, j \in [n]$.

Step 2: For each $i \in [n] \setminus A_0$, do the following:

Step 2-1: Define $b_i := \min\{\hat{a}_{ij} \mid j \in [n] \setminus \{i\}\} - \alpha$.

Step 2-2: Update $\hat{a}_{ij} \leftarrow \hat{a}_{ij} - b_i$ for $j \in [n] \setminus \{i\}$.

Step 3: If $(\hat{a}_{ij})_{i,j\in[n]}$ satisfies

$$\hat{a}_{ij} = \begin{cases} \alpha_{pq} & \text{if } i \in A_p \text{ and } j \in A_q, \\ \alpha_{(i,p)} & \text{if } i \in A_0 \text{ and } j \in A_p \end{cases}$$

for some α_{pq} (distinct $p, q \in [m]$) and some $\alpha_{(i,p)}$ ($p \in [m]$), output that "f is M-convex." Otherwise, output that "f is not M-convex." □

Algorithm III (for Type III).

Step 1: Define $\alpha_{pq} := \min\{a_{ij} \mid i \in A_p, j \in A_q\}$ for distinct $p, q \in [m]$, $b_{(i,q)} := \min\{a_{ij} \mid j \in A_q\} - \alpha_{pq}$ for $p \in [m] \setminus \{q\}$ and $i \in A_p$, and $\hat{a}_{ij} := a_{ij}$ for distinct $i, j \in [n]$.

Step 2: Update $\hat{a}_{ij} \leftarrow \hat{a}_{ij} - b_{(i,q)} - b_{(j,p)}$ for distinct $p, q \in [m]$, $i \in A_p$, and $j \in A_q$.

Step 3: If $(\hat{a}_{ij})_{i,j\in[n]}$ satisfies $\hat{a}_{ij} = \alpha_{pq}$ for all distinct $p, q \in [m]$, $i \in A_p$, and $j \in A_q$, output that "f is M-convex." Otherwise, output that "f is not M-convex." □

Theorem 11 *Algorithms* I, II, *and* III *work correctly and run in* $O(n^2)$ *time.*

By Theorem 11, we obtain Theorem 3. The proofs of the validity of Algorithms I, II, III had to be omitted due to space constraints. They will be included in the full version of this paper.

It is clear that the running time of Algorithms II and III are $O(n^2)$. In the rest of this paper, we see that the running time of Algorithm I is $O(n^2)$. In particular, we can determine whether $(\hat{a}_{ij})_{i,j\in[n]}$ satisfies the anti-ultrametric property in $O(n^2)$ time. First we present a key lemma for designing an $O(n^2)$-time algorithm.

Lemma 12 ([5, Lemma 8]) $(\hat{a}_{ij})_{i,j\in[n]}$ *satisfies the anti-ultrametric property if and only if there exist some laminar family* $\mathcal{L}$ *on* $[n]$ *and some* $c_U \in \overline{\mathbf{R}}$ *for* $U \in \mathcal{L}$ *such that*

- $[n] \in \mathcal{L}$,
- *if* $U \subsetneq U'$, *then* $c_U > c_{U'}$ *holds,*
- $\hat{a}_{ij} = c_{U(i,j)}$ *holds for any distinct* $i, j \in [n]$, *where* $U(i,j)$ *is the minimal element in* $\mathcal{L}$ *including* $\{i, j\}$.

By Lemma 12, we obtain the following natural procedure `Decompose`, which updates a laminar family $\mathcal{L}$ and defines $c_U \in \overline{\mathbf{R}}$ for $U \in \mathcal{L}$. Suppose that we are given $U \subseteq [n]$ and $w \in \overline{\mathbf{R}}$.

Procedure: `Decompose`(U, w)**.**

Step 1: If $|U| \leq 1$ or $w = +\infty$, then stop.

Step 2: Take any $i \in U$. Define $e := \min\{\hat{a}_{ij} \mid j \in U \setminus \{i\}\}$ and $X := \operatorname{argmin}\{\hat{a}_{ij} \mid j \in U \setminus \{i\}\}$.

Step 3: If $e > w$, then $\mathcal{L} \leftarrow \mathcal{L} \cup \{U\}$, $c_U := e$, and $w \leftarrow e$.

Step 4: Execute `Decompose`(X, w) and `Decompose`$(U \setminus X, w)$. □

For initialization, let $\mathcal{L} := \{[n]\}$ and $c_{[n]} := \alpha$. Observe that if $(\hat{a}_{ij})_{i,j\in[n]}$ satisfies the anti-ultrametric property, `Decompose`$([n], \alpha)$ constructs the appropriate laminar family $\mathcal{L}$ and c_U for $U \in \mathcal{L}$ corresponding to $(\hat{a}_{ij})_{i,j\in[n]}$. Moreover `Decompose`$([n], \alpha)$ runs in $O(n^2)$ time.

We are ready to describe an algorithm for checking the anti-ultrametric property as follows.

Algorithm I′ (for checking the anti-ultrametric property).

Step 1: Define $\mathcal{L} := \{[n]\}$ and $c_{[n]} := \alpha$.

Step 2: Execute `Decompose`$([n], \alpha)$.

Step 3: Make a copy of $\mathcal{L}$ and denote it by $\mathcal{L}'$, that is, $\mathcal{L}' := \{U' \mid U \in \mathcal{L}\}$ (the base set of $\mathcal{L}'$ is also $[n]$).

Step 4: While $\mathcal{L}' \neq \emptyset$, do the following:

> **Step 4-1:** Take any minimal element $U' \in \mathcal{L}'$. Define $a'_{ij} := c_U$ for $\{i, j\} \subseteq U$ and $\{i, j\} \cap U' \neq \emptyset$.
>
> **Step 4-2:** Let $U_+ \in \mathcal{L}$ be the minimal element in $\mathcal{L}$ with $U \subsetneq U_+$. Update $U'_+ \leftarrow U'_+ \setminus U$.
>
> **Step 4-3:** Update $\mathcal{L}' \leftarrow \mathcal{L}' \setminus U'$.

Step 5: If $(\hat{a}_{ij})_{i,j\in[n]} = (a'_{ij})_{i,j\in[n]}$, then output "$(\hat{a}_{ij})_{i,j\in[n]}$ satisfies the anti-ultrametric property." Otherwise, output "$(\hat{a}_{ij})_{i,j\in[n]}$ does not satisfy the anti-ultrametric property." □

In Step 4 of Algorithm I′, note that we define the value of a'_{ij} exactly once for every distinct $i, j \in [n]$. Hence the time complexity of Step 4 is $O(n^2)$ time. Thus, we see that Algorithm I′ runs in $O(n^2)$ time. By Lemma 12, the validity of Algorithm I′ is clear. Therefore we obtain the following theorem.

Theorem 13 *Algorithm* I′ *works correctly and runs in* $O(n^2)$ *time.*

By Theorem 13, we can determine whether $(\hat{a}_{ij})_{i,j\in[n]}$ satisfies the anti-ultrametric property in $O(n^2)$ time.

Remark 14 The procedure `Decompose` has already been proposed in the preprint version of [4] in the context of M-convexity, and [1, 12] in the context of ultrametrics. However, these papers considered the restricted case where a_{ij} takes a finite value for all distinct $i, j \in [n]$.

Acknowledgments

We thank Hiroshi Hirai and Kazuo Murota for careful reading and numerous helpful comments.

References

[1] J. C. Colberson and P. Rudnicki. A fast algorithm for construction trees from distance matrices. *Information Processing Letters*, 30:215–220, 1989.

[2] A. W. M. Dress and W. Wenzel. Valuated matroids: A new look at the greedy algorithm. *Applied Mathematics Letters*, 3(2):33–35, 1990.

[3] A. W. M. Dress and W. Wenzel. Valuated matroids. *Advances in Mathematics*, 93:214–250, 1992.

[4] H. Hirai and K. Murota. M-convex functions and tree metrics. *Japan Journal of Industrial and Applied Mathematics*, 21:391–403, 2004. (preprint version: *Mathematical Engineering Technical Reports*, METR 2003-23, University of Tokyo, 2003.)

[5] Y. Iwamasa, K. Murota, and S. Živný. Discrete convexity in joint winner property. arXiv:1701.07645v1, 2017.

[6] K. Murota. Convexity and Steinitz's exchange property. *Advances in Mathematics*, 124:272–311, 1996.

[7] K. Murota. *Discrete Convex Analysis*. SIAM, Philadelphia, 2003.

[8] K. Murota. Recent developments in discrete convex analysis. In W. Cook, L. Lovász, and J. Vygen, editors, *Research Trends in Combinatorial Optimization*, chapter 11, pages 219–260. Springer-Verlag, Berlin, 2009.

[9] K. Murota. Discrete convex analysis: A tool for economics and game theory. *Journal of Mechanism and Institution Design*, 1(1):151–273, 2016.

[10] K. Murota and A. Shioura. Quadratic M-convex and L-convex functions. *Advances in Applied Mathematics*, 33:318–341, 2004.

[11] C. Semple and M. Steel. *Phylogenetics*. Oxford University Press, Oxford, 2003.

[12] M. S. Waterman, T. F. Smith, M. Singh, and W. A. Beyer. Additive evolutionary trees. *Journal of Theoretical Biology*, 64:199–213, 1977.

Index Reduction via Unimodular Transformations[1]

Satoru Iwata

Department of Mathematical Informatics
University of Tokyo
Tokyo 113-8656, Japan
iwata@mist.i.u-tokyo.ac.jp

Mizuyo Takamatsu

Faculty of Science and Engineering
Chuo University
Tokyo 112-8551, Japan
takamatsu@ise.chuo-u.ac.jp

Abstract: This paper presents an algorithm for transforming a matrix pencil $A(s)$ into another matrix pencil $U(s)A(s)$ with a unimodular matrix $U(s)$ so that the resulting Kronecker index is at most one. The algorithm is based on the framework of combinatorial relaxation, which combines graph-algorithmic techniques and matrix computation. Our algorithm works for index reduction of linear differential-algebraic equations, including those for which the existing index reduction methods based on Pantelides' algorithm are known to fail.

Keywords: matrix pencil, index reduction, bipartite matching, combinatorial relaxation, differential-algebraic equations

1 Introduction

A matrix pencil is a polynomial matrix in which the degree of each entry is at most one. By a strict equivalence transformation, each matrix pencil can be brought into its Kronecker canonical form (KCF). Numerically stable computation of KCF is a challenging problem, which has required enormous efforts [2, 4, 5, 10, 21].

Let $A(s)$ be an $n \times n$ matrix pencil. The Kronecker index $\nu(A)$ of $A(s)$ is defined in terms of the KCF of $A(s)$. Previous work given in [7, 8, 14, 20] aims at finding $\nu(A)$ without obtaining the KCF. They utilize the following combinatorial characterization:

$$\nu(A) = \delta_{n-1}(A) - \delta_n(A) + 1. \tag{1}$$

Here, $\delta_k(A)$ denotes the maximum degree of minors of order k in $A(s)$, i.e.,

$$\delta_k(A) = \max\{\deg \det A(s)[I, J] \mid |I| = |J| = k\}, \tag{2}$$

where $\deg a(s)$ designates the degree of a polynomial $a(s)$ and $A(s)[I, J]$ denotes the submatrix with row set I and column set J.

While the previous work [7, 8, 14, 20] deals with the index computation, this paper focuses on the index reduction of a matrix pencil. Our aim is to transform $A(s)$ into another matrix pencil with the Kronecker index at most one. More precisely, we present an algorithm for finding a unimodular polynomial matrix $U(s)$ such that $U(s)A(s)$ is a matrix pencil with $\nu(UA) \leq 1$.

Once the KCF of $A(s)$ is obtained together with the transformation matrices, it is straightforward to construct such a unimodular matrix $U(s)$. Since numerical difficulty is inherent in the computation of KCF, we aim at finding $U(s)$ more directly without relying on the KCF. Instead of computing the KCF, our algorithm makes use of (1). It is known that the value of $\delta_n(A)$ is invariant under unimodular equivalence transformations, which indicates $\delta_n(UA) = \delta_n(A)$. On the other hand, $\delta_{n-1}(UA) = \delta_{n-1}(A)$ does not hold in general. In order to achieve $\nu(UA) \leq 1$, we find $U(s)$ satisfying $\delta_{n-1}(UA) \leq \delta_n(UA) = \delta_n(A)$.

[1]This work was supported by JST CREST, Grant Number JPMJCR14D2, Japan.

Our motivation comes from the study of differential-algebraic equations (DAEs) [1, 3, 6, 11, 19]. Consider a linear DAE

$$F\frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} + H\boldsymbol{z}(t) = \boldsymbol{g}(t) \tag{3}$$

with an initial condition $\boldsymbol{z}(0) = \boldsymbol{z}_0$, where F and H are constant matrices. By the Laplace transformation, we obtain

$$A(s)\tilde{\boldsymbol{z}}(s) = \tilde{\boldsymbol{g}}(s) + F\boldsymbol{z}_0$$

with the matrix pencil $A(s) = sF + H$. The numerical difficulty of the DAE (3) is measured by the Kronecker index $\nu(A)$.

A common approach for solving a high index DAE is to transform it into an equivalent DAE with index at most one, which can be solved easily by numerical methods including the backward differentiation formulas (BDF). This motivates a variety of index reduction algorithms, in which we are allowed to differentiate a certain equation and add it to another equation. Such an operation corresponds to equivalence row transformations with unimodular polynomial matrix $U(s)$. The Laplace transform of the resulting DAE is in the form of

$$U(s)A(s)\tilde{\boldsymbol{z}}(s) = U(s)(\tilde{\boldsymbol{g}}(s) + F\boldsymbol{z}_0).$$

If $U(s)A(s)$ is a matrix pencil and $\nu(UA) \leq 1$ holds, the index of the DAE is now reduced to at most one.

The modeling and simulation software for dynamical systems, such as Dymola, OpenModelica, and MapleSim, is equipped with the index reduction methods based on Pantelides' algorithm [17], the dummy derivative approach [12], or the signature method [18]. These algorithms adopt a structural approach, which extracts zero/nonzero pattern of coefficients in equations, ignoring the numerical values. Such algorithms are efficient, because they exploit graph-algorithmic techniques. However, the discard of numerical information can cause a failure even for linear DAEs. In contrast, our algorithm always works for any instances of linear DAEs.

The algorithms for computing $\delta_k(A)$ given in [7, 8, 14, 20] are based on the framework of "combinatorial relaxation," which combines graph-algorithmic techniques and matrix computation. The combinatorial relaxation approach is invented by Murota [13] for computing the Newton diagram of Puiseux-series solutions to determinantal equations and then applied to the computation of the degree of determinants of polynomial matrices [15]. In combinatorial relaxation algorithms for computing $\delta_k(A)$, we find an estimate $\hat{\delta}_k(A)$ of $\delta_k(A)$ by solving a matching problem and check if $\hat{\delta}_k(A) = \delta_k(A)$ by constant matrix computation. If $\hat{\delta}_k(A) \neq \delta_k(A)$, then we modify $A(s)$ to improve $\hat{\delta}_k(A)$ without changing $\delta_k(A)$. After a finite number of iterations, the algorithms terminate with $\hat{\delta}_k(A) = \delta_k(A)$. They mainly rely on fast combinatorial algorithms and perform numerical computation only when necessary.

Our index reduction algorithm, which consists of two phases, inherits the idea of combinatorial relaxation. In the first phase, we transform $A(s)$ into another matrix pencil $\tilde{A}(s)$ such that an estimate of $\nu(\tilde{A})$ is at most one. In the second phase, we determine if the estimate is correct. If not, we further transform $\tilde{A}(s)$ into another matrix pencil $\hat{A}(s)$ with $\nu(\hat{A}) \leq 1$. In both phases, we exploit a feasible dual solution of the matching problem, which was also used by Pryce [18] in the interpretation of Pantelides' algorithm [17].

The rest of this paper is organized as follows. In Section 2, we explain the bipartite matching problems associated with matrix pencils. We present an index reduction algorithm in Section 3. Section 4 gives numerical examples, and Section 5 concludes this paper. Due to the space limitation, we omit proofs in this note. The readers are referred to [9] for more technical details.

2 Matrix Pencils and Matching Problems

For a polynomial $a(s)$, we denote the degree of $a(s)$ by $\deg a$, where $\deg 0 = -\infty$ by convention. A polynomial matrix $A(s) = (a_{ij}(s))$ with $\deg a_{ij} \leq 1$ for all (i, j) is called a *matrix pencil*. A matrix pencil $A(s)$ is said to be *regular* if $A(s)$ is square and $\det A(s)$ is a nonvanishing polynomial.

Let us denote by block-diag$(D_1, \ldots, D_b)$ the block-diagonal matrix pencil with diagonal blocks $D_1, \ldots, D_b$. By a strict equivalence transformation, a regular matrix pencil $A(s)$ can be brought into its Kronecker canonical form block-diag$(sI_{\mu_0} + J_{\mu_0}, N_{\mu_1}, \ldots, N_{\mu_d})$, where I_{μ_0} is a $\mu_0 \times \mu_0$ identity matrix, J_{μ_0} is a $\mu_0 \times \mu_0$ constant matrix, and N_μ is a $\mu \times \mu$ matrix pencil defined by

$$N_\mu = \begin{pmatrix} 1 & s & 0 & \cdots & 0 \\ 0 & 1 & s & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & s \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}.$$

The matrices $N_{\mu_1}, \ldots, N_{\mu_d}$ are called the *nilpotent blocks*.

For a matrix pencil $A(s)$, the Kronecker index $\nu(A)$ is defined to be the maximum size of the nilpotent blocks in the Kronecker canonical form of $A(s)$, i.e., $\max_{1 \le i \le d} \mu_i$. It is known [16, Theorem 5.1.8] that $\nu(A)$ is expressed by (1).

A polynomial matrix is called *unimodular* if it is square and its determinant is a nonvanishing constant. This implies that a square polynomial matrix is unimodular if and only if its inverse is a polynomial matrix.

Let $A(s)$ be an $n \times n$ regular matrix pencil with row set R and column set C. We construct a bipartite graph $G(A) = (R, C; E(A))$ with $E(A) = \{(i,j) \mid i \in R, j \in C, A_{ij}(s) \neq 0\}$. The weight c_e of an edge $e = (i,j)$ is given by $c_e = c_{ij} = \deg A_{ij}(s)$. We remark that c_e is 0 or 1 for each $e \in E(A)$ because $A(s)$ is a matrix pencil. A subset M of $E(A)$ is called a *matching* if every pair of edges in M is disjoint. A matching M is called a *perfect matching* if M covers all the vertices.

Consider the following maximum-weight perfect matching problem P(A):

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in M} c_e \\ \text{subject to} \quad & M \text{ is a perfect matching.} \end{aligned}$$

Since $A(s)$ is regular, $G(A)$ has a perfect matching. The maximum weight of a perfect matching in $G(A)$, denoted by $\hat{\delta}_n(A)$, is an upper bound on $\delta_n(A)$.

The dual problem D(A) of P(A) is given by

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in R} p_i - \sum_{j \in C} q_j \\ \text{subject to} \quad & p_i - q_j \ge c_e \quad (e = (i,j) \in E(A)), \\ & p_i \in \mathbb{Z} \quad (i \in R), \\ & q_j \in \mathbb{Z} \quad (j \in C). \end{aligned}$$

We denote the objective function of D(A) by $\Delta_n(p, q)$.

We construct an optimal solution (p, q) of D(A) as follows. Let M be a maximum-weight perfect matching in $G(A) = (R, C; E(A))$. The reorientation of $a \in E(A)$ is denoted by $\bar{a}$. Consider an auxiliary graph $\check{G}_M = (\check{V}, \check{E})$ with $\check{V} = R \cup C \cup \{r\}$ and $\check{E} = \bar{E} \cup M \cup W$, where r is a new vertex, $\bar{E} = \{\bar{a} \mid a \in E(A)\}$, and $W = \{(r,i) \mid i \in R\}$. We define the arc length $\gamma : \check{E} \to \mathbb{Z}$ by

$$\gamma(a) = \begin{cases} -c_{\bar{a}} & (a \in \bar{E}) \\ c_a & (a \in M) \\ 0 & (a \in W) \end{cases}.$$

Let $d(i,j)$ be the shortest distance from $i \in \check{V}$ to $j \in \check{V}$ with respect to the arc length γ in $\check{G}_M$. We define

$$p_i = -d(r,i) + \max_{\ell \in C} d(r,\ell) \quad (i \in R), \tag{4}$$

$$q_j = -d(r,j) + \max_{\ell \in C} d(r,\ell) \quad (j \in C). \tag{5}$$

Lemma 1 *Suppose that* (p,q) *is defined by* (4) *and* (5). *Then* (p,q) *is an optimal solution of* $\mathrm{D}(A)$ *satisfying*

$$\min_{i \in R} p_i \geq 0, \quad \min_{j \in C} q_j = 0, \quad \max_{j \in C} q_j \leq n. \tag{6}$$

Next, consider the following matching problem corresponding to $\delta_{n-1}(A)$.

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in M} c_e \\ \text{subject to} \quad & M \text{ is a matching}, \\ & |M| = n-1. \end{aligned}$$

The optimal value is denoted by $\hat{\delta}_{n-1}(A)$, which is an upper bound on $\delta_{n-1}(A)$.

For a feasible solution (p,q) of $\mathrm{D}(A)$, we define

$$\Delta_{n-1}(p,q) = \Delta_n(p,q) - \min_{i \in R} p_i + \max_{j \in C} q_j.$$

The following lemma gives upper bounds on $\hat{\delta}_n(A)$ and $\hat{\delta}_{n-1}(A)$.

Lemma 2 *For a feasible solution* (p,q) *of* $\mathrm{D}(A)$, *we have*

$$\hat{\delta}_n(A) \leq \Delta_n(p,q), \quad \hat{\delta}_{n-1}(A) \leq \Delta_{n-1}(p,q).$$

3 Index Reduction Algorithm

3.1 Outline of Algorithm

Let $A(s)$ be an $n \times n$ regular matrix pencil, and (p,q) be a feasible solution of $\mathrm{D}(A)$ satisfying (6). By Lemma 2, we have

$$\delta_n(A) \leq \hat{\delta}_n(A) \leq \Delta_n(p,q), \tag{7}$$

$$\delta_{n-1}(A) \leq \hat{\delta}_{n-1}(A) \leq \Delta_{n-1}(p,q). \tag{8}$$

Our aim is to find a unimodular matrix $U(s)$ such that $\bar{A}(s) = U(s)A(s)$ is a matrix pencil with index $\nu(\bar{A}) \leq 1$. The following algorithm updates a matrix pencil $A(s)$ and a feasible solution (p,q). The upper bounds $\Delta_n(p,q)$ and $\Delta_{n-1}(p,q)$ are non-increasing. The resulting matrix pencil $\bar{A}(s)$ and its feasible solution $(\bar{p},\bar{q})$ satisfy

$$\delta_n(\bar{A}) = \hat{\delta}_n(\bar{A}) = \Delta_n(\bar{p},\bar{q}), \quad \delta_{n-1}(\bar{A}) = \hat{\delta}_{n-1}(\bar{A}) = \Delta_{n-1}(\bar{p},\bar{q}), \tag{9}$$

$$\bar{p}_i \in \{0,1\} \quad (i \in R), \qquad \bar{q}_j = 0 \quad (j \in C). \tag{10}$$

We describe the outline of the index reduction algorithm. The algorithm consists of two phases. In the first phase, we make use of

$$\hat{\nu}(p,q) := \Delta_{n-1}(p,q) - \Delta_n(p,q) + 1$$

as an estimate of $\nu(A) = \delta_{n-1}(A) - \delta_n(A) + 1$. At the end of the first phase, we obtain an updated matrix pencil $A(s)$ and a feasible solution (p, q) with $\hat{\nu}(p, q) \leq 1$. It should be remarked that this does not imply $\nu(A) \leq 1$, because $\hat{\nu}(p, q)$ is not an upper bound on $\nu(A)$.

In the second phase, we check if both $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p, q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p, q)$ hold without computing $\delta_n(A)$ and $\delta_{n-1}(A)$ directly. If these equations hold, we obtain

$$\nu(A) = \delta_{n-1}(A) - \delta_n(A) + 1 = \Delta_{n-1}(p, q) - \Delta_n(p, q) + 1 = \hat{\nu}(p, q) \leq 1.$$

If not, we further update $A(s)$ to another matrix pencil. A formal description is as follows.

Outline of Index Reduction Algorithm

Step 1: Construct an optimal solution (p, q) of $\mathrm{D}(A)$ satisfying (6).

Step 2: If $q_j = 0$ for every $j \in C$, then go to Step 4.

Step 3: Bring $A(s)$ into another matrix pencil $\tilde{A}(s)$ by a unimodular transformation, and construct a feasible solution $(\tilde{p}, \tilde{q})$ of $\mathrm{D}(\tilde{A})$ from (p, q). Set $A(s) \leftarrow \tilde{A}(s)$ and $(p, q) \leftarrow (\tilde{p}, \tilde{q})$. Go back to Step 2.

Step 4: If both $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p, q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p, q)$ hold, then terminate.

Step 5: Bring $A(s)$ into another matrix pencil $\hat{A}(s)$ by a unimodular transformation, and construct a feasible solution $(\hat{p}, \hat{q})$ of $\mathrm{D}(\hat{A})$ from (p, q). Set $A(s) \leftarrow \hat{A}(s)$ and $(p, q) \leftarrow (\hat{p}, \hat{q})$. Go back to Step 4.

Phase 1 corresponds to Steps 1–3, while Phase 2 corresponds to Steps 4–5. In Steps 1–3, we aim at constructing a feasible solution (p, q) satisfying (10), which implies $\hat{\nu}(p, q) \leq 1$. Then we further update p to obtain a feasible solution satisfying (9) in Steps 4–5. The details of Steps 3–5 are given in Sections 3.2–3.4, respectively.

3.2 Unimodular Transformations in Step 3

We describe how to construct $(\tilde{p}, \tilde{q})$ from a feasible solution (p, q) of $\mathrm{D}(A)$ satisfying (6) in Step 3. For nonnegative integer h, we define

$$R_h = \{i \in R \mid p_i = h\}, \quad C_h = \{j \in C \mid q_j = h\}.$$

Then $A(s)$ is expressed as

$$A(s) = \begin{array}{c} \\ R_\eta \\ R_{\eta-1} \\ \vdots \\ \vdots \\ R_1 \\ R_0 \end{array} \begin{array}{c} \begin{array}{cccccc} C_\eta & C_{\eta-1} & C_{\eta-2} & \cdots & C_1 & C_0 \end{array} \\ \begin{pmatrix} * & ** & ** & \cdots & \cdots & ** \\ O & * & ** & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & ** \\ O & \cdots & \cdots & O & * & ** \\ O & \cdots & \cdots & O & O & * \end{pmatrix} \end{array}$$

for some η, where $*$ and $**$ denote a constant matrix and a matrix pencil, respectively. Since $A(s)$ is regular, the submatrix $A[R_0, C_0]$ is of full-row rank, and hence we can express it as $\begin{pmatrix} * & H_0 \end{pmatrix}$ with a nonsingular constant matrix H_0.

Next, consider the submatrix

$$\begin{array}{c} \\ R_1 \\ R_0 \end{array} \begin{array}{c} \begin{array}{c} C_0 \end{array} \\ \begin{pmatrix} ** & sF_1 + H_1 \\ * & H_0 \end{pmatrix} \end{array}$$

with constant matrices F_1 and H_1. By multiplying a unimodular matrix $\begin{pmatrix} I & -sF_1H_0^{-1} \\ O & I \end{pmatrix}$ from the left, we obtain

$$\begin{array}{c} \\ R_1 \\ R_0 \end{array} \begin{array}{c} C_0 \\ \begin{pmatrix} sF_2+H_2 & H_1 \\ * & H_0 \end{pmatrix} \end{array}$$

with constant matrices F_2 and H_2. Since $A[R_0, C_1] = O$, this transformation does not change $A[R_1, C_1]$.

Then consider the submatrix $\left(\begin{array}{c|c} sF_2+H_2 & H_1 \end{array}\right)$, which can be transformed into

$$\left(\begin{array}{cc|c} sF_3+H_3 & ** & * \\ * & * & * \end{array}\right)$$

by row transformations, so that the lower part does not contain s with nonsingular constant matrix F_3 and constant matrix H_3.

As a result, we obtain another matrix pencil $\tilde{A}(s)$ satisfying the following conditions.

- It holds that
$$\tilde{A}(s)[R_1 \cup R_0, C_1 \cup C_0] = \left(\begin{array}{c|cc|c} * & sF_3+H_3 & ** & * \\ * & * & * & * \\ \hline O & * & * & H_0 \end{array}\right), \tag{11}$$
where the first two row sets correspond to R_1, the last row set corresponds to R_0, the first column set corresponds to C_1, and the last three column sets correspond to C_0.
- The other entries coincide with the corresponding entries of $A(s)$.

Let us denote the first row set of (11) by S. We construct $(\tilde{p}, \tilde{q})$ from (p, q) by

$$\begin{aligned} &\tilde{p}_i = p_i - 1 \quad (i \in R \setminus (R_0 \cup S)), && \tilde{p}_i = p_i \quad (i \in R_0 \cup S), \\ &\tilde{q}_j = q_j - 1 \quad (j \in C \setminus C_0), && \tilde{q}_j = q_j = 0 \quad (j \in C_0). \end{aligned}$$

The following lemma ensures that $(\tilde{p}, \tilde{q})$ is a feasible solution of $\mathrm{D}(\tilde{A})$.

Lemma 3 *Let (p, q) be a feasible solution of $\mathrm{D}(A)$ satisfying (6). Then $(\tilde{p}, \tilde{q})$ is a feasible solution of $D(\tilde{A})$ satisfying (6).*

The following lemma shows that the values of the right-hand sides in (7) and (8) decrease or remain the same when we update (p, q) to $(\tilde{p}, \tilde{q})$.

Lemma 4 *Let (p, q) be a feasible solution of $\mathrm{D}(A)$ satisfying (6). The dual solution $(\tilde{p}, \tilde{q})$ obtained by the above procedure satisfies*

$$\Delta_n(p, q) \geq \Delta_n(\tilde{p}, \tilde{q}), \quad \Delta_{n-1}(p, q) \geq \Delta_{n-1}(\tilde{p}, \tilde{q}).$$

By executing Steps 1–3, we obtain a matrix pencil $A(s)$ and its feasible solution (p, q) with the following property.

Lemma 5 *At the end of Phase 1, we obtain (p, q) such that $p_i \in \{0, 1\}$ for every $i \in R$ and $q_j = 0$ for every $j \in C$. Moreover, the number of iterations in Phase 1 is at most n.*

Lemma 5 leads to the following corollary.

Corollary 6 *At the end of Phase 1, we have $\hat{\nu}(p, q) \leq 1$.*

3.3 Test for Tightness in Step 4

In this section, we present how to check if both $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p,q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p,q)$ hold in Step 4.

Suppose that we have a feasible solution (p,q) of $\mathrm{D}(A)$ such that $p_i \in \{0,1\}$ for every $i \in R$ and $q_j = 0$ for every $j \in C$. The tight coefficient matrix of $A(s)$ is defined to be the constant matrix $A^{\#} = (A^{\#}_{ij})$ with $A^{\#}_{ij}$ being the coefficient of $s^{p_i - q_j}$ in $A_{ij}(s)$. The following lemma enables us to check $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p,q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p,q)$ efficiently.

Lemma 7 *The tight coefficient matrix $A^{\#}$ is nonsingular if and only if both $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p,q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p,q)$ hold.*

By Lemma 7, we can perform Step 4 by checking the nonsingularity of $A^{\#}$.

3.4 Unimodular Transformations in Step 5

Let $A(s)$ be a matrix pencil in Step 5. The algorithm has detected that the condition in Step 4 is not fulfilled, i.e., the tight coefficient matrix $A^{\#}$ is singular. Hence there exists a nonzero row vector $\boldsymbol{u} = (u_i \mid i \in R)$ such that

$$\boldsymbol{u}A^{\#} = \boldsymbol{0}.$$

By executing the Gaussian elimination on $A^{\#}$ with column transformations, we can find $\boldsymbol{u}$ such that $\operatorname{supp} \boldsymbol{u} := \{i \in R \mid u_i \neq 0\}$ is minimal with respect to set inclusion.

By the definition of $A^{\#}$, we have $A^{\#}[R_0, C] = A(s)[R_0, C]$. Since $A(s)$ is regular, $A^{\#}[R_0, C]$ is of full-row rank. This implies that there exists $l \in \operatorname{supp} \boldsymbol{u}$ with $p_l = 1$.

We now define U by

$$U_{ik} = \begin{cases} u_k/u_l & (i = l), \\ \delta_{ik} & (i \neq l), \end{cases}$$

where δ_{ik} denotes Kronecker's delta. We remark that the row set and the column set of U correspond to $R_1 \cup R_0$ and $U[R_0, R_1] = O$. We denote by $\operatorname{diag}(s;p)$ the square diagonal matrix with each (i,i) entry being s^{p_i}. Then the polynomial matrix $U(s) = \operatorname{diag}(s;p) \cdot U \cdot \operatorname{diag}(s;-p)$ is unimodular.

Since $A(s)$ can be expressed as

$$A(s) = \operatorname{diag}(s;p) \cdot \left(A^{\#} + \frac{1}{s} \begin{pmatrix} A(0)[R_1, C] \\ O \end{pmatrix} \right),$$

it holds that

$$\begin{aligned} U(s)A(s) &= \operatorname{diag}(s;p) \cdot U \cdot \left(A^{\#} + \frac{1}{s} \begin{pmatrix} * \\ O \end{pmatrix} \right) = \operatorname{diag}(s;p) \cdot \left(UA^{\#} + \frac{1}{s} \begin{pmatrix} * & * \\ O & * \end{pmatrix} \begin{pmatrix} * \\ O \end{pmatrix} \right) \\ &= \operatorname{diag}(s;p) \cdot \left(UA^{\#} + \frac{1}{s} \begin{pmatrix} * \\ O \end{pmatrix} \right) = \operatorname{diag}(s;p) \cdot UA^{\#} + \begin{pmatrix} * \\ O \end{pmatrix}, \end{aligned}$$

where $*$ denotes a constant matrix. Hence $U(s)A(s)$ remains to be a matrix pencil. Since the lth row vector of $UA^{\#}$ is zero, $U(s)A(s)$ does not contain s in the lth row. Hence we can decrease $p_l = 1$ by one. By setting

$$\hat{A}(s) := U(s)A(s), \quad \hat{p}_i := \begin{cases} 0 & (i = l), \\ p_i & (i \neq l), \end{cases} \quad \hat{q} := q,$$

we obtain another matrix pencil $\hat{A}(s)$ and its feasible solution $(\hat{p}, \hat{q})$.

Lemma 8 *The number of iterations in Phase 2 is at most n.*

At the end of the index reduction algorithm, we obtain a matrix pencil with index at most one.

Theorem 9 *The algorithm finds a matrix pencil with the Kronecker index at most one in $\mathrm{O}(n^4)$ time.*

4 Examples

We give two examples below.

Example 10 The following is a famous example for which Pantelides' algorithm does not work:

$$\begin{aligned} z_1 - \dot{z}_1 + 2z_2 + 3z_3 &= 0, \\ z_1 + z_2 + z_3 + 1 &= 0, \\ 2z_1 + z_2 + z_3 &= 0. \end{aligned}$$

The corresponding matrix pencil $A(s)$ is expressed as $A(s) = \begin{pmatrix} -s+1 & 2 & 3 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix}$. By $\delta_2(A) = 1$ and $\delta_3(A) = 0$, the index $\nu(A)$ is equal to 2. However, when we apply Pantelides' algorithm [17] to $A(s)$, the algorithm terminates without detecting equations to be differentiated. Pantelides' algorithm is adopted in the MATLAB function called `reduceDAEIndex`. In fact, this function does not work for the DAE.

Let us apply our algorithm to $A(s)$. In Step 1, we find an optimal solution $p = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ and $q = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}$ of $\mathrm{D}(A)$. In Step 3, we obtain another solution $p = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$ and $q = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$ without changing $A(s)$. Then we go to Step 4 by $q = \mathbf{0}$. The tight coefficient matrix $A^{\#} = \begin{pmatrix} -1 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix}$ is singular. In Step 5, we have $\boldsymbol{u} = \begin{pmatrix} 1 & -1 & 1 \end{pmatrix}$ and $U(s) = \begin{pmatrix} 1 & -s & s \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. The matrix pencil $A(s)$ is transformed into $U(s)A(s) = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix}$ with $p = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$ and $q = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$. Then we obtain $\nu(UA) = 1$.

Example 11 Consider another matrix pencil

$$A(s) = \begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & s \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & s \end{pmatrix}.$$

It follows from $\delta_3(A) = 2$ and $\delta_4(A) = 0$ that $\nu(A) = 3$. We apply the algorithm described in Section 3 to $A(s)$.

In Step 1, we find an optimal solution $p = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$ and $q = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}$ of $\mathrm{D}(A)$. Then we go to Step 3 by $q \neq \mathbf{0}$. In Step 3, we delete s in the last row by row transformations and obtain a feasible dual solution $p' = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}$ and $q' = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$ as follows:

$$A(s) = \begin{array}{c} \\ R_1 \end{array} \begin{array}{c} \begin{matrix} C_1 & C_0 \end{matrix} \\ \begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & s \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & s \end{pmatrix} \end{array} \quad \longrightarrow \quad A'(s) = U^{\circ}(s)A(s) = \begin{array}{c} \\ R_1 \\ \\ R_0 \end{array} \begin{array}{c} \begin{matrix} C_0 \end{matrix} \\ \begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & s \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array},$$

where $U^{\circ}(s) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}$. We return to Step 2 and then go to Step 4 by $q' \neq \mathbf{0}$. The tight

coefficient matrix $A^{\#} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$ is singular in Step 4, and we have $\boldsymbol{u}' = \begin{pmatrix} 0 & 1 & -1 & 1 \end{pmatrix}$ and $U'(s) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -s & s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ in Step 5. The matrix pencil $A'(s)$ is transformed into

$$A''(s) = U'(s)A'(s) = \begin{array}{c} \\ R_1 \\ R_0 \end{array} \begin{array}{c} C_0 \\ \begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

with $p'' = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}$ and $q'' = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$.

Returning to Step 4, the tight coefficient matrix $A^{\#} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$ is also singular. In Step 5, we have $\boldsymbol{u}'' = \begin{pmatrix} 1 & -1 & 0 & 0 \end{pmatrix}$ and $U''(s) = \begin{pmatrix} 1 & -s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$. The matrix pencil $A''(s)$ is transformed into

$$\bar{A}(s) = U''(s)A''(s) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

with $\bar{p} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$ and $\bar{q} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$. Returning to Step 4, the tight coefficient matrix $A^{\#} = \bar{A}(s)$ is nonsingular and hence we terminate the algorithm.

As a result, we obtain a unimodular matrix $U(s)$ and a matrix pencil $\bar{A}(s)$ with $\nu(\bar{A}) = 1$ expressed as

$$U(s) = U''(s)U'(s)U^{\circ}(s) = \begin{pmatrix} 1 & s^2 - s & s^2 & -s^2 \\ 0 & -s+1 & -s & s \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}, \quad \bar{A}(s) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

5 Conclusion

We have presented a new index reduction algorithm of matrix pencils which makes use of unimodular transformations. The algorithm is based on the framework of combinatorial relaxation, which combines graph-algorithmic techniques and matrix computation. Our algorithm can be used as an index reduction method for linear DAEs. It works correctly for any linear DAEs including those for which Pantelides' algorithm is known to fail. An extension of our algorithm to index reduction of nonlinear DAEs is left for future investigation.

References

[1] U. M. Ascher and L. R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Philadelphia, 1998.

[2] T. Beelen and P. Van Dooren, *An improved algorithm for the computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 105 (1988), pp. 9–65.

[3] K. E. Brenan, S. L. Campbell and L. R. Petzold, Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, SIAM, Philadelphia, 2nd edition, 1996.

[4] J. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil A-λB: Robust software with error bounds and applications. Part I: Theory and algorithms*, ACM Trans. Math. Softw., 19 (1993), pp. 160–174.

[5] J. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil A-λB: Robust software with error bounds and applications. Part II: Software and applications*, ACM Trans. Math. Softw., 19 (1993), pp. 175–201.

[6] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II, Springer-Verlag, Berlin, 2nd edition, 1996.

[7] S. Iwata, *Computing the maximum degree of minors in matrix pencils via combinatorial relaxation*, Algorithmica, 36 (2003), pp. 331–341.

[8] S. Iwata, K. Murota, and I. Sakuta, *Primal-dual combinatorial relaxation algorithms for the maximum degree of subdeterminants*, SIAM J. Sci. Comput., 17 (1996), pp. 993–1012.

[9] S. Iwata and M. Takamatsu, *Index reduction via unimodular transformations*, METR 2017-05, Department of Mathematical Informatics, University of Tokyo, 2017.

[10] B. Kågström, *RGSVD—an algorithm for computing the Kronecker structure and reducing subspaces of singular $A - \lambda B$ pencils*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 185–211.

[11] P. Kunkel and V. Mehrmann, Differential-Algebraic Equations: Analysis and Numerical Solutions, European Mathematical Society, Zürich, 2006.

[12] S. E. Mattsson and G. Söderlind, *Index reduction in differential-algebraic equations using dummy derivatives*, SIAM J. Sci. Comput., 14 (1993), pp. 677–692.

[13] K. Murota, *Computing Puiseux-series solutions to determinantal equations via combinatorial relaxation*, SIAM J. Comput., 19 (1990), pp. 1132–1161.

[14] K. Murota, *Combinatorial relaxation algorithm for the maximum degree of subdeterminants: Computing Smith-McMillan form at infinity and structural indices in Kronecker form*, Appl. Algebra Engrg. Comm. Comput., 6 (1995), pp. 251–273.

[15] K. Murota, *Computing the degree of determinants via combinatorial relaxation*, SIAM J. Comput., 24 (1995), pp. 765–796.

[16] K. Murota, Matrices and Matroids for Systems Analysis, Springer-Verlag, Berlin, 2000.

[17] C. C. Pantelides, *The consistent initialization of differential-algebraic systems*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 213–231.

[18] J. D. Pryce, *A simple structural analysis method for DAEs*, BIT, 41 (2001), pp. 364–394.

[19] R. Riaza, Differential-Algebraic Systems: Analytical Aspects and Circuit Applications, World Scientific Publishing Company, Singapore, 2008.

[20] S. Sato, *Combinatorial relaxation algorithm for the entire sequence of the maximum degree of minors*, Algorithmica, 77 (2017), pp. 815–835.

[21] P. Van Dooren, *The computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 27 (1979), pp. 103–140.

List Supermodular Coloring[1]

Satoru Iwata[2]

Department of Mathematical Informatics
University of Tokyo
Tokyo 113-8654, Japan
iwata@mist.i.u-tokyo.ac.jp

Yu Yokoi[3]

Department of Mathematical Informatics
University of Tokyo
Tokyo 113-8654, Japan
yu_yokoi@mist.i.u-tokyo.ac.jp

Abstract: In 1995, Galvin provided an elegant proof for the list edge coloring conjecture for bipartite graphs, utilizing the stable matching theorem of Gale and Shapley. In this paper, we generalize Galvin's result to the setting of supermodular coloring, introduced by Schrijver, with the aid of the monochromatic path theorem of Sands, Sauer and Woodrow.

Keywords: List coloring, Intersecting supermodular functions.

1 Introduction

A list coloring is a type of coloring in which each of the elements to be colored has its own list of permissible colors. One of the most celebrated results in the study of list coloring is the following theorem of Galvin on edge colorings of bipartite graphs. An *edge coloring* of an undirected graph is a function which assigns a color to each edge so that no two adjacent edges have the same color.

Theorem 1 (Galvin [8]) *For a bipartite graph that admits an edge coloring with $k \in \mathbf{Z}_{>0}$ colors, if each edge e has a list $L(e)$ of k colors, then there exists an edge coloring such that every edge e is assigned a color in $L(e)$.* ■

The existence of an edge coloring in a bipartite graph is characterized by Kőnig's theorem.

Theorem 2 (Kőnig [9]) *A bipartite graph admits an edge coloring with k or less colors if and only if each vertex is incident to at most k edges.* ■

In other words, the minimum number of colors required for a bipartite edge coloring is equal to the maximum degree. Combining Theorems 1 and 2 we see that, if the size of $L(e)$ for each edge e is at least the maximum degree, there is an edge coloring which assigns a color in $L(e)$ for each edge e.

In this paper, we generalize the above result of Galvin to the setting of supermodular coloring introduced by Schrijver [11]. Let U be a finite set. We say that $X, Y \subseteq U$ are *intersecting* if none of $X \cap Y$, $X \setminus Y$ and $Y \setminus X$ are empty. A family $\mathcal{F} \subseteq 2^U$ is called an *intersecting family* if every intersecting pair of $X, Y \in \mathcal{F}$ satisfies $X \cup Y, X \cap Y \in \mathcal{F}$. A function $g\colon \mathcal{F} \to \mathbf{R}$ is called *intersecting-supermodular* if $\mathcal{F}$ is an intersecting family and g satisfies the *supermodular inequality* $g(X) + g(Y) \leq g(X \cup Y) + g(X \cap Y)$ for every intersecting pair of $X, Y \in \mathcal{F}$.

For any $k \in \mathbf{Z}_{>0}$, we write $[k] := \{1, 2, \ldots, k\}$. A function $\pi : U \to [k]$ is called a k-coloring. We say that π *dominates* a function $g : \mathcal{F} \to \mathbf{Z}$ if $|\pi(X)| \geq g(X)$ holds for every $X \in \mathcal{F}$, where $\pi(X) := \{\, \pi(u) \mid u \in X \,\}$. For two intersecting-supermodular functions $g_1 : \mathcal{F}_1 \to \mathbf{Z}$ and $g_2 : \mathcal{F}_2 \to \mathbf{Z}$, a k-coloring π is called a *supermodular k-coloring* if π dominates both g_1 and g_2. Schrijver characterized the existence of a supermodular k-coloring in the following theorem, which generalizes Theorem 2.

[1]This work is supported by CREST, Japan Science and Technology Agency.
[2]Supported by MEXT Grant-in-Aid for Scientific Research on Innovative Areas (No. 24106005).
[3]Supported by JSPS Fellowship for Young Scientists.

Theorem 3 (Schrijver [11]) *Let $g_1 : \mathcal{F}_1 \to \mathbf{Z}$ and $g_2 : \mathcal{F}_2 \to \mathbf{Z}$ be intersecting-supermodular functions such that each g_i satisfies $|X| \geq g_i(X)$ for every $X \in \mathcal{F}_i$. Then, for any $k \in \mathbf{Z}_{>0}$, there exists a supermodular k-coloring for (g_1, g_2) if and only if both $k \geq \max\{\, g_1(X) \mid X \in \mathcal{F}_1 \,\}$ and $k \geq \max\{\, g_2(X) \mid X \in \mathcal{F}_2 \,\}$ hold.* ■

Tardos [12] provided an alternative proof for this theorem using the generalized matroid intersection theorem. Theorem 3 has been extended to skew-supermodular coloring [5] and to a further general framework [6].

Let us consider the list coloring version of supermodular colorings. Let Σ be a finite set of colors and let each $u \in U$ have a color list $L(u) \subseteq \Sigma$, that is, L is a mapping from U to 2^{Σ}. For intersecting-supermodular functions $g_1 : \mathcal{F}_1 \to \mathbf{Z}$, $g_2 : \mathcal{F}_2 \to \mathbf{Z}$ and color lists $\{L(u)\}_{u \in U}$, a *list supermodular coloring* is a function $\varphi : U \to \Sigma$ such that every $u \in U$ satisfies $\varphi(u) \in L(u)$ and φ dominates both g_1 and g_2. The main result of this paper is as follows.

Theorem 4 *For intersecting-supermodular functions $g_1 : \mathcal{F}_1 \to \mathbf{Z}$ and $g_2 : \mathcal{F}_2 \to \mathbf{Z}$ and $k \in \mathbf{Z}_{>0}$, assume that there exists a supermodular k-coloring for (g_1, g_2). If L satisfies $|L(u)| = k$ for each $u \in U$, then there exists a list supermodular coloring $\varphi : U \to \Sigma$ for (g_1, g_2, L).* ■

The pair (g_1, g_2) of intersecting-supermodular functions is called *k-choosable* if, for every $L : U \to 2^{\Sigma}$ with $|L(u)| = k$ $(\forall u \in U)$, there exists a list supermodular coloring for (g_1, g_2, L). Combining Theorems 3 and 4 implies the following corollary.

Corollary 5 *Let $g_1 : \mathcal{F}_1 \to \mathbf{Z}$ and $g_2 : \mathcal{F}_2 \to \mathbf{Z}$ be intersecting-supermodular functions such that each g_i satisfies $|X| \geq g_i(X)$ for every $X \in \mathcal{F}_i$. Then, for any $k \in \mathbf{Z}_{>0}$, the pair (g_1, g_2) is k-choosable if and only if both $k \geq \max\{\, g_1(X) \mid X \in \mathcal{F}_1 \,\}$ and $k \geq \max\{\, g_2(X) \mid X \in \mathcal{F}_2 \,\}$ hold.* ■

A surprising aspect of Galvin's proof is that it utilizes a famous result of Gale and Shapley [7] on the existence of stable matchings in bipartite graphs. See also [1] for a beautiful exposition. To show Theorem 4, we utilize the monochromatic path theorem of Sands, Sauer and Woodrow [10]. This theorem states the existence of a kernel for a pair of posets, and was shown by Fleiner [3] to be a generalization of the result of Gale and Shapley.

The rest of this paper is organized as follows. In Section 2, we introduce the monochromatic path theorem of Sands et al. [10]. In Section 3, we introduce skeleton posets for colorings that dominate intersecting-supermodular functions. The existence proof of skeleton posets is postponed to Section 5. In Section 4, we give a proof of Theorem 4 using induction on the ground set. Each step of the induction applies the monochromatic path theorem to skeleton posets. Section 6 shows the skew-supermodular extension of Theorem 4, whose statement has been conjectured in [2].

2 Monochromatic Path Theorem

As mentioned in Introduction, a key ingredient of our proof of Theorem 4 is to use the monochromatic path theorem of Sands, Sauer and Woodrow [10]. Here we introduce the theorem with the terminology of Fleiner and Jankó [4].

In a partially ordered set (poset) $P = (U, \preceq)$, two elements $u, v \in U$ are *comparable* if $u \preceq v$ or $v \preceq u$ holds, and otherwise they are *incomparable*. A *chain* is a subset in which each pair of elements is comparable. An *antichain* is a subset in which each pair of distinct elements is incomparable. Let $P_1 = (U, \preceq_1)$ and $P_2 = (U, \preceq_2)$ be two posets on the same ground set U. A subset $K \subseteq U$ is called a *kernel* if K is a common antichain and every element $u \in U \setminus K$ admits an element $v \in K$ such that $v \prec_1 u$ or $v \prec_2 u$. Moreover, for any subset $S \subseteq U$, we call its subset $K \subseteq S$ a *kernel of S* if K is a common antichain and every element $u \in S \setminus K$ admits an element $v \in K$ such that $v \prec_1 u$ or $v \prec_2 u$. We are now ready to describe the theorem of Sands et al.

Theorem 6 (Sands et al. [10]) *Let $P_1 = (U, \preceq_1)$ and $P_2 = (U, \preceq_2)$ be posets on the same ground set U. For any subset $S \subseteq U$, there exists a kernel of S.*

The original statement of Sands et al. was described in terms of directed graphs whose edges are colored with two colors, and the binary relation $v \prec u$ in Theorem 6 corresponds to the existence of a monochromatic path from a node u to another node v. Their statement can be applied to more general binary relations but only to the case of $S = U$. It is easy to see the equivalence between their original statement and Theorem 6.

3 Skeleton Posets

Let $g : \mathcal{F} \to \mathbf{Z}$ be an intersecting-supermodular function on $\mathcal{F} \subseteq 2^U$. For a subset $K \subseteq U$, the *reduction* of g by K is the function $g_K : \mathcal{F}_K \to \mathbf{Z}$ defined by $\mathcal{F}_K = \{ Z \setminus K \mid Z \in \mathcal{F} \}$ and

$$g_K(X) = \max \{ \hat{g}_K(Z) \mid Z \in \mathcal{F},\ Z \setminus K = X \} \quad (X \in \mathcal{F}_K),$$

where $\hat{g}_K : \mathcal{F} \to \mathbf{Z}$ is defined by

$$\hat{g}_K(Z) = \begin{cases} g(Z) - 1 & (Z \in \mathcal{F},\ Z \cap K \neq \emptyset), \\ g(Z) & (Z \in \mathcal{F},\ Z \cap K = \emptyset). \end{cases}$$

Claim 7 *The reduction $g_K : \mathcal{F}_K \to \mathbf{Z}$ is an intersecting-supermodular function.*

Proof: For every $X, Y \in \mathcal{F}_K$ and $Z_X, Z_Y \in \mathcal{F}$ such that $Z_X \setminus K = X$ and $Z_Y \setminus K = Y$, we have $X \cup Y = (Z_X \setminus K) \cup (Z_X \setminus K) = (Z_X \cup Z_Y) \setminus K$, and the same holds for the intersection. These imply that $\mathcal{F}_K$ is an intersecting family. We now show the supermodular inequality of g_K for intersecting $X, Y \in \mathcal{F}_K$. Take $Z_X, Z_Y \in \mathcal{F}$ which attain $g_K(X) = \hat{g}_K(Z_X)$ and $g_K(Y) = \hat{g}_K(Z_Y)$. As easily confirmed, $\hat{g}_K : \mathcal{F} \to \mathbf{Z}$ is intersecting supermodular, and hence $\hat{g}_K(Z_X) + \hat{g}_K(Z_Y) \leq \hat{g}_K(Z_X \cup Z_Y) + \hat{g}_K(Z_X \cap Z_Y)$. Also, we have $\hat{g}_K(Z_X \cup Z_Y) \leq g_K(X \cup Y)$ because $Z_X \cup Z_Y \in \mathcal{F}$ and $(Z_X \cup Z_Y) \setminus K = X \cup Y$. Similarly, $\hat{g}_K(Z_X \cap Z_Y) \leq g_K(X \cap Y)$ holds. Combining these inequalities, we obtain $g_K(X) + g_K(Y) \leq g_K(X \cup Y) + g_K(X \cap Y)$. □

Let $\pi : U \to [k]$ be a k-coloring. We say that a poset $P = (U, \preceq)$ is *consistent* with π if $u \prec v$ implies $\pi(u) < \pi(v)$ for every $u, v \in U$. For a consistent poset P and a subset $K \subseteq U$, the *reduction* of π by K in P is the k-coloring $\pi_K : U \setminus K \to [k]$ defined by

$$\pi_K(u) = \begin{cases} \pi(u) - 1 & (\exists v \in K : v \prec u), \\ \pi(u) & (\text{otherwise}). \end{cases}$$

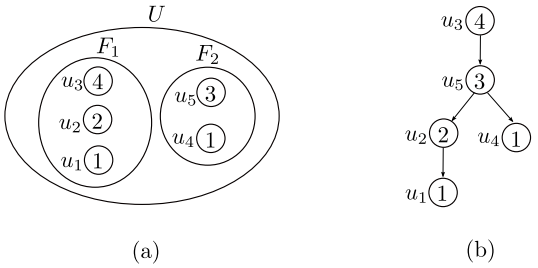
Note that every $u \in U \setminus K$ indeed satisfies $\pi_K(u) \geq 1$ because of the consistency of P.

Definition 8 *A* skeleton poset *of (π, g) is a poset $P = (U, \preceq)$ which is consistent with π and satisfies the following condition: For every antichain K in P, the reduction of π by K in P dominates the reduction of g by K.* ■

Here, we provide a sufficient condition for the existence of a skeleton poset. We call π a *g-dominating k-coloring* if it dominates g. A g-dominating k-coloring π is called *minimal* if there is no g-dominating k-coloring $\tilde{\pi} : U \to [k]$ such that $\tilde{\pi}(u) < \pi(u)$ for some $u \in U$ and $\tilde{\pi}(v) = \pi(v)$ for every other $v \in U \setminus \{u\}$.

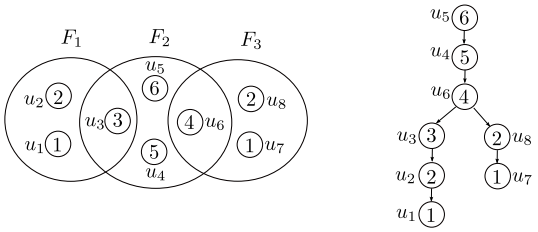
Proposition 9 *For every intersecting-supermodular function $g : \mathcal{F} \to \mathbf{Z}$ and every minimal g-dominating k-coloring $\pi : U \to [k]$, there exists a skeleton poset P of (π, g).* ■

The proof of Proposition 9 is postponed to Section 5. Instead, we demonstrate some examples of skeleton posets below.



(a) (b)

Figure 1: (a) The family $\mathcal{F}$ in Example 10. The value $\pi(u)$ of each $u \in U$ is written in the circle corresponding to u. (b) The Hasse diagram which defines a skeleton poset of (π, g) in Example 10.



(a) (b)

Figure 2: (a) The family $\mathcal{F}$ in Example 11. The value $\pi(u)$ of each $u \in U$ is written in the circle corresponding to u. (b) The Hasse diagram which defines a skeleton poset of (π, g) in Example 11.

Example 10 *Let $U = \{u_1, u_2, u_3, u_4, u_5\}$ and $\mathcal{F} = \{F_1, F_2, U\}$, where $F_1 = \{u_1, u_2, u_3\}$, $F_2 = \{u_4, u_5\}$. Define $g : \mathcal{F} \to \mathbf{Z}$ by $g(F_1) = 3$, $g(F_2) = 2$, $g(U) = 4$ and $\pi : U \to [k]$ by $(\pi(u_1), \pi(u_2), \ldots, \pi(u_5)) = (1, 2, 4, 1, 3)$, where $k \geq 4$ (see Figure 1 (a)). Then, π is a minimal g-dominating k-coloring. Let $P = (U, \preceq)$ be a poset whose Hasse diagram is depicted in Figure 1 (b). We can check that P is a skeleton poset of (π, g).*

Example 11 *Let $U = \{u_1, u_2, \ldots, u_8\}$. Let $F_1 = \{u_1, u_2, u_3\}$, $F_2 = \{u_3, u_4, u_5, u_6\}$, $F_3 = \{u_6, u_7, u_8\}$ and $\mathcal{F} = \{F_1, F_2, F_3, F_1 \cap F_2, F_2 \cap F_3, F_1 \cup F_2, F_2 \cup F_3, U\}$. Define $g : \mathcal{F} \to \mathbf{Z}$ by $g(F_1) = 3$, $g(F_2) = 2$, $g(F_3) = 3$, $g(F_1 \cap F_2) = g(F_2 \cap F_3) = 1$, $g(F_1 \cup F_2) = g(F_2 \cup F_3) = 4$, and $g(U) = 6$. Define $\pi : U \to [k]$ by $(\pi(u_1), \pi(u_2), \ldots, \pi(u_8)) = (1, 2, 3, 5, 6, 4, 1, 2)$, where $k \geq 6$ (see Figure 2 (a)). Then, π is a minimal g-dominating k-coloring. Let $P = (U, \preceq)$ be a poset whose Hasse diagram is depicted in Figure 2 (b). We see that P is a skeleton poset of (π, g).*

4 Proof

In this section, we give a proof to Theorem 4 relying on Theorem 6 and Proposition 9. Let $g_1 : \mathcal{F}_1 \to \mathbf{Z}$ and $g_2 : \mathcal{F}_2 \to \mathbf{Z}$ be intersecting-supermodular functions on $\mathcal{F}_1, \mathcal{F}_2 \subseteq 2^U$ and let $k \in \mathbf{Z}_{>0}$.

Lemma 12 *If $\pi_1, \pi_2 : U \to [k]$ dominate g_1 and g_2, respectively, then for any nonempty subset $S \subseteq U$, there exist nonempty $K \subseteq S$ and k-colorings $\pi_1', \pi_2' : U \setminus K \to [k]$ that satisfy the following conditions.*

(a) *For every $u \in U \setminus K$, we have $\pi_1'(u) + \pi_2'(u) \leq \pi_1(u) + \pi_2(u)$.*
Moreover, $u \in S \setminus K$ implies $\pi_1'(u) + \pi_2'(u) < \pi_1(u) + \pi_2(u)$.

(b) *For each $i \in \{1, 2\}$, π_i' dominates the reduction of g_i by K.*

PROOF: For each $i \in \{1,2\}$, since π_i dominates g_i, there is a minimal g_i-dominating k-coloring $\hat{\pi}_i : U \to [k]$ with $\hat{\pi}_i \le \pi_i$. By Proposition 9, there is a skeleton poset $P_i = (U, \preceq_i)$ of $(\hat{\pi}_i, g_i)$ for each i. Take any nonempty $S \subseteq U$ and apply Theorem 6 to P_1, P_2, and S. Then, we obtain a kernel K of S. That is, $K \subseteq S$ is a common antichain and every $u \in S \setminus K$ admits some $v \in K$ such that $v \prec_1 u$ or $v \prec_2 u$. Let $\pi'_i : U \setminus K \to [k]$ be the reduction of $\hat{\pi}_i$ by K in P_i. Then $\pi'_1(u) + \pi'_2(u) \le \hat{\pi}_1(u) + \hat{\pi}_2(u)$ for every $u \in U \setminus K$ and strict inequality holds for every $u \in S \setminus K$ by the definition of a kernel. As $\hat{\pi}_1(u) \le \pi_1(u)$, $\hat{\pi}_2(u) \le \pi_2(u)$ for every $u \in U$, condition (a) follows. Since P_i is a skeleton poset and K is an antichain in P_i for each $i \in \{1,2\}$, π'_i dominates the reduction of g_i by K. □

Recall that Σ is a set of colors and $L : U \to 2^\Sigma$ is an assignment of color lists to elements.

Proposition 13 *For $L : U \to 2^\Sigma$, assume that there exist k-colorings $\pi_1, \pi_2 : U \to [k]$ satisfying the following conditions.*

(i) *For every $u \in U$, we have $\pi_1(u) + \pi_2(u) - 1 \le |L(u)|$.*

(ii) *For each $i \in \{1,2\}$, π_i dominates g_i.*

Then there exists a list supermodular coloring $\varphi : U \to \Sigma$ for (g_1, g_2, L).

PROOF: We show this by induction on $|U|$. If $|U| = 1$, the statement is obvious.

If $|U| > 1$, take some $l \in \bigcup \{ L(u) \mid u \in U \}$ and let $S := \{ u \in U \mid l \in L(u) \}$. By Lemma 12, there exist nonempty $K \subseteq S$ and $\pi'_1, \pi'_2 : U \setminus K \to [k]$ satisfying (a) and (b). For each $i \in \{1,2\}$, let g'_i denote the reduction of g_i by K. Then, g'_i is intersecting supermodular by Claim 7, and π'_i dominates g'_i by (b). Define $L' : U \setminus K \to 2^\Sigma$ by $L'(u) = L(u) \setminus \{l\}$ for each $u \in U \setminus K$. It then follows from (a) that $\pi'_1(u) + \pi'_2(u) - 1 \le |L'(u)|$ for every $u \in U \setminus K$. Thus, π'_1, π'_2 satisfy (i) and (ii) with $(U \setminus K, g'_1, g'_2, L')$ in place of (U, g_1, g_2, L). By the inductive assumption, there exists a list supermodular coloring $\varphi' : U \setminus K \to \Sigma$ for (g'_1, g'_2, L'). Define $\varphi : U \to \Sigma$ by

$$\varphi(u) = \begin{cases} \varphi'(u) & (u \in U \setminus K), \\ l & (u \in K). \end{cases}$$

Then, clearly $\varphi(u) \in L(u)$ for every $u \in U$. For each $i \in \{1,2\}$, every $X \in \mathcal{F}_i$ with $X \cap K \neq \emptyset$ satisfies $|\varphi(X)| = |\varphi'(X \setminus K)| + 1 \ge g'_i(X \setminus K) + 1 \ge g_i(X)$, and every $X \in \mathcal{F}_i$ with $X \cap K = \emptyset$ satisfies $|\varphi(X)| = |\varphi'(X \setminus K)| \ge g'_i(X \setminus K) \ge g_i(X)$. Thus φ is a list supermodular coloring for (g_1, g_2, L). □

PROOF OF THEOREM 4: Recall that L satisfies $|L(u)| = k$ for every $u \in U$. Also, we are provided a supermodular k-coloring $\pi : U \to [k]$ which dominates both g_1 and g_2. Let $\pi_1(u) := \pi(u)$ and $\pi_2(u) := k+1-\pi(u)$ for every $u \in U$. They satisfy the condition (i) of Proposition 13 as $\pi_1(u)+\pi_2(u)-1 = k = |L(u)|$ for every u. Also (ii) holds as $g_i(X) \le |\pi(X)| = |\pi_1(X)| = |\pi_2(X)|$ for every $i \in \{1,2\}$ and $X \in \mathcal{F}_i$. Proposition 13 then implies the statement of Theorem 4. □

5 Existence of Skeleton Posets

Let $g : \mathcal{F} \to \mathbf{Z}$ be an intersecting-supermodular function and $\pi : U \to [k]$ be a minimal g-dominating k-coloring. In this section, we prove Proposition 9 by constructing a skeleton poset $P = (U, \preceq)$ of (π, g). We first define the poset and then show that it is indeed a skeleton poset of (π, g).

5.1 Poset Construction

We call a subset $X \in \mathcal{F}$ *tight* if $|\pi(X)| = g(X)$ holds. Note that the function $|\pi(\cdot)| : 2^U \to \mathbf{Z}$ is *submodular*, that is, $|\pi(X)| + |\pi(Y)| \ge |\pi(X \cup Y)| + |\pi(X \cap Y)|$ for any $X, Y \subseteq U$. This implies the following fact.

Claim 14 *If $X, Y \in \mathcal{F}$ are tight and intersecting, then $X \cup Y$, $X \cap Y \in \mathcal{F}$ are also tight.*

PROOF: Since $|\pi(\cdot)|$ is submodular and π dominates g, we have

$$g(X) + g(Y) = |\pi(X)| + |\pi(Y)| \geq |\pi(X \cup Y)| + |\pi(X \cap Y)| \geq g(X \cup Y) + g(X \cap Y).$$

As g is intersecting-supermodular, $g(X \cup Y) + g(X \cap Y) \geq g(X) + g(Y)$ also holds. Then, all the above inequalities are in fact equalities and we obtain $|\pi(X \cup Y)| = g(X \cup Y)$ and $|\pi(X \cap Y)| = g(X \cap Y)$. □

Claim 15 *If $X, Y \in \mathcal{F}$ are tight and intersecting, then $\pi(X) \cap \pi(Y) = \pi(X \cap Y)$.*

PROOF: Clearly, $\pi(X \cap Y) \subseteq \pi(X) \cap \pi(Y)$. We then show $|\pi(X \cap Y)| = |\pi(X) \cap \pi(Y)|$ to complete the proof. As shown in the proof of Claim 14, $|\pi(X)| + |\pi(Y)| = |\pi(X \cup Y)| + |\pi(X \cap Y)|$. Also, we see $\pi(X) \cup \pi(Y) = \pi(X \cup Y)$. These imply $|\pi(X \cap Y)| = |\pi(X)| + |\pi(Y)| - |\pi(X \cup Y)| = |\pi(X)| + |\pi(Y)| - |\pi(X) \cup \pi(Y)| = |\pi(X) \cap \pi(Y)|$. □

Claim 16 *For any $u \in U$ with $\pi(u) > 1$ and $j \in \{1, \ldots, \pi(u) - 1\}$, there exists $F_j \in \mathcal{F}$ which satisfies $u \in F_j$, $|\pi(F_j)| = g(F_j)$, $\pi(F_j - u) \not\ni \pi(u)$, and $\pi(F_j) \ni j$.*

PROOF: Let $\pi' : U \to [k]$ be a k-coloring such that $\pi'(u) = j$ and $\pi'(v) = \pi(v)$ for every $v \in U \setminus \{u\}$. Since π is a minimal g-dominating k-coloring, π' does not dominate g. Hence there exists F_j such that $|\pi'(F_j)| < g(F_j)$. As $|\pi(F_j)| \geq g(F_j)$ holds, we have $|\pi'(F_j)| < |\pi(F_j)|$, which implies the four conditions in the statement. □

Claim 17 *For any $u \in U$ with $\pi(u) > 1$, there exist one or more $F \in \mathcal{F}$ such that*

$$u \in F, \tag{1}$$
$$|\pi(F)| = g(F), \tag{2}$$
$$\pi(F - u) \not\ni \pi(u), \tag{3}$$
$$\pi(F) \supseteq \{1, 2, \ldots, \pi(u)\}. \tag{4}$$

Furthermore, among all such $F \in \mathcal{F}$, there exists a unique minimal one.

PROOF: For each $j \in \{1, 2, \ldots, \pi(u) - 1\}$, let $F_j \in \mathcal{F}$ be a subset which satisfies the four conditions in Claim 16. Then $F := \bigcup \{ F_j \mid j = 1, 2, \ldots, \pi(u) - 1 \}$ satisfies (1)–(4). Condition (2) follows from Claim 14 since all F_j contain u. The other three are clear by definition. To show the existence of the minimum, we show that, if both F and F' satisfy (1)–(4), then so does $F \cap F'$. By definition, (1) and (3) are clear. Claims 14 and 15 imply (2) and (4), respectively. □

For any $u \in U$ with $\pi(u) > 1$, denote by $D(u)$ the unique minimal $F \in \mathcal{F}$ satisfying (1)–(4). For $u \in U$ with $\pi(u) = 1$, let $D(u)$ be $\{u\}$. Define $\prec$ by

$$u \prec v \iff [D(u) \subseteq D(v),\ \pi(u) < \pi(v)]$$

and let $u \preceq v$ mean $u \prec v$ or $u = v$. Let $P = (U, \preceq)$. Then, P is a poset consistent with π.

Claim 18 *If $D(u) \cap D(v) \neq \emptyset$, then u and v are comparable.*

PROOF: Let $u \neq v$ since otherwise the claim is obvious. We assume $\pi(u) \leq \pi(v)$ without loss of generality.

In the case $\pi(u) = 1$, we have $D(u) = \{u\} \subseteq D(v)$ since $D(u) \cap D(v) \neq \emptyset$. As $u \neq v$, then $D(v) \neq \{v\}$, which implies $\pi(v) > 1 = \pi(u)$, and hence $u \prec v$.

In the case $\pi(u) > 1$, we have $D(u), D(v) \in \mathcal{F}$. Since $D(u)$ and $D(v)$ are tight and intersecting, Claims 14 and 15 imply that $D(u) \cap D(v)$ is tight and satisfies $\pi(D(u) \cap D(v)) \supseteq \{1, 2, \ldots, \pi(u)\}$. The latter implies $D(u) \cap D(v) \ni u$ since $D(u)$ satisfies $\pi(D(u) - u) \not\ni \pi(u)$. Thus, conditions (1)–(4) hold

for u and $F = D(u) \cap D(v)$. By the minimality of $D(u)$, this implies $D(u) \cap D(v) = D(u)$, and hence $D(u) \subseteq D(v)$. Also, as $D(v)$ satisfies $\pi(D(v) - v) \not\ni \pi(v)$, the condition $u \in D(u) \subseteq D(v)$ implies $\pi(u) \neq \pi(v)$, and hence $\pi(u) < \pi(v)$. Thus, $u \prec v$ holds. □

By Claim 18, $D(u) \cap D(v) \neq \emptyset$ implies $D(u) \subseteq D(v)$ or $D(u) \supseteq D(v)$, i.e., $\{ D(u) \mid u \in U \}$ forms a *laminar family*.

Claim 19 *For any $u \in U$ with $\pi(u) > 1$, there exists $v \in U$ with $\pi(v) = \pi(u) - 1$ and $v \prec u$.*

Proof: Since (4) holds with $F = D(u)$, there is $v \in D(u)$ with $\pi(v) = \pi(u) - 1$. As $v \in D(v) \cap D(u) \neq \emptyset$ and $\pi(v) < \pi(u)$, Claim 18 implies $v \prec u$. □

Claim 20 *If $v \preceq u$, then $v \in D(u)$. Conversely, if $v \in D(u)$, then $v \preceq u$ or $u \prec v$.*

Proof: The condition $v \preceq u$ implies $v \in D(v) \subseteq D(u)$, and the first claim holds. Also, $v \in D(u)$ implies $v \in D(v) \cap D(u) \neq \emptyset$, and hence v is comparable with u by Claim 18. □

Claim 21 *If $u \preceq v$ and $u \preceq w$, then v and w are comparable.*

Proof: Since $D(u) \subseteq D(v) \cap D(w) \neq \emptyset$, Claim 18 implies the statement. □

Claim 21 implies that the Hasse diagram of $P = (U, \preceq)$ forms a *branching*, i.e., a collection of rooted directed trees.

Claim 22 *For each $u \in U$ with $\pi(u) > 1$, let $C(u)$ be a longest chain included in $D(u)$. Then, the following statements hold:*

- $\pi(C(u)) \supseteq \{1, 2, \ldots, \pi(u)\}$,
- $\pi(D(u) \setminus C(u)) \subseteq \{1, 2, \ldots, \pi(u) - 1\}$,
- $\pi(C(u)) = \pi(D(u))$ *and* $g(D(u)) = |C(u)|$.

Proof: By Claims 20 and 21, $D_{\mathrm{up}}(u) := \{ v \in D(u) \mid \pi(v) \geq \pi(u) \}$ forms a chain whose minimum is u. Also, every element v in $D_{\mathrm{down}}(u) := \{ v \in D(u) \mid \pi(v) \leq \pi(u) \}$ satisfies $v \preceq u$ by Claim 20, and hence any longest chain in $D_{\mathrm{down}}(u)$ contains u as the maximum. As $C(u)$ is a longest chain in $D(u)$, it satisfies $C(u) = D_{\mathrm{up}}(u) \cup C_{\mathrm{down}}$ for some longest chain C_{down} in $D_{\mathrm{down}}(u)$.

By Claim 19, there is a chain $v_{\pi(u)} \succ v_{\pi(u)-1} \succ \cdots \succ v_1$ such that $v_{\pi(u)} = u$ and $\pi(v_j) = j$ for each $j = 1, 2, \ldots, \pi(u)$. Each v_j belongs to $D_{\mathrm{down}}(u)$ by Claim 20, and this chain is longest in $D_{\mathrm{down}}(u)$ because its elements have all possible values of π in $D_{\mathrm{down}}(u)$. Since C_{down} is also a longest chain in $D_{\mathrm{down}}(u)$, it has the same length $\pi(u)$, and hence $\pi(C_{\mathrm{down}}) = \{1, 2, \ldots, \pi(u)\}$.

The first statement follows from $C(u) = D_{\mathrm{up}}(u) \cup C_{\mathrm{down}} \supseteq C_{\mathrm{down}}$. The second one follows from $D(u) \setminus C(u) = D(u) \setminus (D_{\mathrm{up}}(u) \cup C_{\mathrm{down}}) \subseteq D_{\mathrm{down}}(u) - u$ and $\pi(D(u) - u) \not\ni \pi(u)$. From the first and second statements, $\pi(C(u)) = \pi(D(u))$ follows. As $C(u)$ is a chain, we have $|C(u)| = |\pi(C(u))| = |\pi(D(u))|$, which equals $g(D(u))$ by the tightness of $D(u)$. □

The following fact will be useful later.

Claim 23 *Assume that $u, v \in U$ satisfies $\pi(v) = \pi(u) - 1$. If $X \in \mathcal{F}$ is tight and $\{u, v\} \subseteq X$ holds, then we have $D(u) \setminus D(v) \subseteq X$.*

Proof: Note that $D(v)$ is a singleton or a member of $\mathcal{F}$. Also, we have $v \in D(v) \cap X \neq \emptyset$. Then $D(v) \cup X$ is a member of $\mathcal{F}$ and tight. As $\pi(u) > 1$, the set $D(u)$ is also in $\mathcal{F}$ and tight. Then, the nonempty set $F := (D(v) \cup X) \cap D(u) \ni u$ is also a member of $\mathcal{F}$ and tight. Note that then conditions (1)–(4) hold for u and F, where (4) follows from Claim 15 and the condition $\pi(v) = \pi(u) - 1$. Therefore, the minimality of $D(u)$ implies $D(u) \subseteq F$, which yields $D(u) \setminus D(v) \subseteq X$. □

5.2 Reduction by an Antichain

We now show that $P = (U, \preceq)$ is indeed a skeleton poset of (π, g). Clearly P is consistent with π, i.e., $u \prec v$ implies $\pi(u) < \pi(v)$. We now show that, for any antichain K in P, the reduction of π by K dominates the reduction g_K of g by K.

Take an antichain $K \subseteq U$. Let $\pi_K : U \setminus K \to [k]$ be the reduction of π by K in P, i.e.,

$$\pi_K(u) = \begin{cases} \pi(u) - 1 & (\exists v \in K : v \prec u), \\ \pi(u) & (\text{otherwise}). \end{cases}$$

To prove that π_K dominates g_K, it suffices to show that $|\pi_K(X \setminus K)| \geq \hat{g}_K(X)$ holds for every $X \in \mathcal{F}$, where $\hat{g}_K : \mathcal{F} \to \mathbf{Z}$ is defined by $\hat{g}_K(X) = g(X) - 1$ for $X \in \mathcal{F}$ with $X \cap K \neq \emptyset$ and $\hat{g}_K(X) = g(X)$ for $X \in \mathcal{F}$ with $X \cap K = \emptyset$.

Claim 24 *For any chain $C \subseteq U$, exactly one of the following holds.*

1. *$|C \cap K| = 1$ and $\pi_K(u) \neq \pi_K(v)$ for every distinct $u, v \in C \setminus K$. Hence $|\pi_K(C \setminus K)| = |C| - 1$.*
2. *$|C \cap K| = 0$ and $\pi_K(u) \neq \pi_K(v)$ for every distinct $u, v \in C \setminus K$. Hence $|\pi_K(C \setminus K)| = |C|$.*
3. *$|C \cap K| = 0$ and just one pair of $u, v \in C$ satisfies $\pi_K(u) = \pi_K(v)$. Hence $|\pi_K(C \setminus K)| = |C| - 1$. If $\pi(v) \leq \pi(u)$ for such $u, v \in C$, then $\pi(v) = \pi(u) - 1$ and $(D(u) \setminus D(v)) \cap K \neq \emptyset$.*

PROOF: The only point to concern is the last statement in the third case. Since C is a chain, the condition $\pi_K(u) = \pi_K(v)$ and the definition of π_K imply $\pi(v) = \pi_K(v) = \pi_K(u) = \pi(u) - 1$. By $\pi_K(u) = \pi(u) - 1$, there exists $w \in K$ with $w \prec u$, which implies $w \in D(u)$ by Claim 20. We now prove $w \notin D(v)$ which completes the proof. Note that $w \prec u$ implies $\pi(w) < \pi(u)$, and hence $\pi(w) \leq \pi(u) - 1 = \pi(v)$. Suppose, to the contrary, $w \in D(v)$. If $\pi(w) = \pi(v)$, then $\pi(D(v) - v) \not\ni \pi(v)$ implies $w = v$, which contradicts $w \in K$, $v \in C$, and $C \cap K = \emptyset$. If $\pi(w) < \pi(v)$, then Claim 20 implies $w \prec v$, which contradicts $\pi_K(v) = \pi(v)$ by the definition of π_K. Thus, we obtain $w \notin D(v)$. □

Lemma 25 *For $X \in \mathcal{F}$, if there is a chain $C \subseteq X$ with $\pi(C) = \pi(X)$, then $|\pi_K(X \setminus K)| \geq \hat{g}_K(X)$.*

PROOF: Since C is a chain and $\pi(C) = \pi(X)$, we have $|C| = |\pi(C)| = |\pi(X)| \geq g(X)$. Let us consider the three cases described in Claim 24.

In the first case, we have $C \cap K \neq \emptyset$ and $|\pi_K(C \setminus K)| = |C| - 1$. Since $X \cap K \supseteq C \cap K \neq \emptyset$ implies $\hat{g}_K(X) = g(X) - 1$, we have $|\pi_K(X \setminus K)| \geq |\pi_K(C \setminus K)| = |C| - 1 \geq g(X) - 1 = \hat{g}_K(X)$.

In the second case, we have $C \cap K = \emptyset$ and $|\pi_K(C \setminus K)| = |C|$, which together with $g(X) \geq \hat{g}_K(X)$ imply $|\pi_K(X \setminus K)| \geq |\pi_K(C \setminus K)| = |C| \geq g(X) \geq \hat{g}_K(X)$.

In the third case, we have $|\pi_K(C \setminus K)| = |C| - 1$, $\pi(v) = \pi(u) - 1$, and $(D(u) \setminus D(v)) \cap K \neq \emptyset$. If X is not tight, then $|C| = |\pi(X)| > g(X)$, and hence $|\pi_K(X \setminus K)| \geq |\pi_K(C \setminus K)| = |C| - 1 \geq g(X) \geq \hat{g}_K(X)$. If X is tight, then Claim 23 and $\pi(v) = \pi(u) - 1$ imply $D(u) \setminus D(v) \subseteq X$. Combined with $(D(u) \setminus D(v)) \cap K \neq \emptyset$, this implies $X \cap K \neq \emptyset$, and hence $\hat{g}_K(X) = g(X) - 1$. Thus, we obtain $|\pi_K(X \setminus K)| \geq |\pi_K(C \setminus K)| = |C| - 1 \geq g(X) - 1 = \hat{g}_K(X)$. □

Lemma 26 *For $u \in U$ with $\pi(u) > 1$, the set $D(u) \in \mathcal{F}$ satisfies $|\pi_K(D(u) \setminus K)| = \hat{g}_K(D(u))$.*

PROOF: By Claim 22, a longest chain $C(u)$ in $D(u)$ satisfies $\pi(C(u)) = \pi(D(u))$ and $|C(u)| = g(D(u))$. Then Lemma 25 implies $|\pi_K(D(u) \setminus K)| \geq \hat{g}_K(D(u))$. We then intend to show $|\pi_K(D(u) \setminus K)| \leq \hat{g}_K(D(u))$.

Claim 22 says $\pi(D(u) \setminus C(u)) \subseteq \{1, 2, \ldots, \pi(u) - 1\}$ and $\pi(C(u)) \supseteq \{1, 2, \ldots, \pi(u)\}$, which imply $\pi_K((D(u) \setminus C(u)) \setminus K) \subseteq \{1, 2, \ldots, \pi(u) - 1\} \subseteq \pi_K(C(u) \setminus K)$ by the definition of π_K. Then, we have $\pi_K(D(u) \setminus K) = \pi_K(C(u) \setminus K)$, which yields $|\pi_K(D(u) \setminus K)| = |\pi_K(C(u) \setminus K)| \leq |C(u)| = g(D(u))$. In particular, if $D(u) \cap K = \emptyset$, then $|\pi_K(D(u) \setminus K)| \leq g(D(u)) = \hat{g}_K(D(u))$.

We now consider the case of $D(u) \cap K \neq \emptyset$. As we have $|\pi_K(D(u) \setminus K)| = |\pi_K(C(u) \setminus K)|$ and $g(D(u)) = |C(u)|$, it suffices to show $|\pi_K(C(u) \setminus K)| < |C(u)|$. If $C(u) \cap K \neq \emptyset$, this is clear. Assume $C(u) \cap K = \emptyset$, and then $D(u) \cap K \neq \emptyset$ implies $(D(u) \setminus C(u)) \cap K \neq \emptyset$. As we have $D(u) \setminus C(u) \subseteq \{ v \in U \mid v \prec u \}$ by Claims 20 and 22, we obtain $\{ v \in U \mid v \prec u \} \cap K \neq \emptyset$, and hence $\pi_K(u) = \pi(u) - 1$. Since $v \preceq u$ implies $\pi_K(v) \leq \pi_K(u)$ for any v, the subset $C' := \{ v \in C(u) \mid v \preceq u \}$ satisfies $\pi_K(C') \subseteq \{1, 2, \ldots, \pi(u) - 1\}$. This implies $|\pi_K(C')| < \pi(u) = |C'|$, where the last equality follows from Claim 22. Therefore, some pair of distinct $v, w \in C' \subseteq C(u)$ satisfies $\pi_K(v) = \pi_K(w)$, and hence $|\pi_K(C(u) \setminus K)| = |\pi_K(C(u))| < |C(u)|$, which is the desired conclusion. □

Proposition 27 *Every $X \in \mathcal{F}$ satisfies $|\pi_K(X \setminus K)| \geq \hat{g}_K(X)$.*

Proof: The proof is by induction with respect to set inclusion.

First, consider the case in which $X \in \mathcal{F}$ is minimal, i.e., there is no $Y \in \mathcal{F}$ with $Y \subsetneq X$. Then, every $u \in X$ with $\pi(u) > 1$ satisfies $X \subseteq D(u)$ since otherwise we have $u \in X \cap D(u) \subsetneq X$ and $X \cap D(u) \in \mathcal{F}$, which contradict the minimality of X. Also $u \in X$ with $\pi(u) = 1$ satisfies $D(u) = \{u\} \subseteq X$. Then, every pair of $u, v \in X$ with $\pi(u) < \pi(v)$ satisfies either $X \subseteq D(u) \cap D(v) \neq \emptyset$ or $\{u\} \subseteq X \subseteq D(v)$. In each case, we have $u \prec v$ by Claim 18. That is, every pair of elements is comparable if their values of π are different. Hence, there is a chain $C \subseteq X$ such that $\pi(C) = \pi(X)$, which implies $|\pi_K(X \setminus K)| \geq \hat{g}_K(X)$ by Lemma 25.

We now intend to show $|\pi_K(X \setminus K)| \geq \hat{g}_K(X)$, assuming inductively that $|\pi_K(Y \setminus K)| \geq \hat{g}_K(Y)$ holds for every $Y \in \mathcal{F}$ with $Y \subsetneq X$.

We start with the case in which every $u \in X$ satisfies $X \subseteq D(u)$. For $u, v \in X$ with $\pi(u) < \pi(v)$, we have $X \subseteq D(u) \cap D(v) \neq \emptyset$, which implies $u \prec v$ by Claim 18. Then, there is a chain $C \subseteq X$ such that $\pi(C) = \pi(X)$, and hence $|\pi_K(X \setminus K)| \geq \hat{g}_K(X)$ by Lemma 25.

We now consider the case in which some $u \in X$ satisfies $X \not\subseteq D(u)$. Among all such elements, let $u \in X$ maximize $\pi(u)$. Then, every $v \in X$ with $\pi(v) > \pi(u)$ satisfies $u \in X \subseteq D(v)$, and hence $v \succ u$ by Claim 20. Recall that every $v \in D(u)$ with $\pi(v) > \pi(u)$ also satisfies $v \succ u$. Then, by Claim 21, $C := \{ v \in X \cup D(u) \mid \pi(v) > \pi(u) \} \cup \{u\}$ forms a chain whose minimum is u. Let $\hat{C}$ be a longest chain subject to $C \subseteq \hat{C} \subseteq X \cup D(u)$. As $\hat{C}$ is longest, Claim 22 implies $\pi(\hat{C}) \supseteq \{1, 2, \ldots, \pi(u)\}$. Therefore, we have $\pi(\hat{C}) \supseteq \pi(C) \cup \{1, 2, \ldots, \pi(u)\} \supseteq \pi(X \cup D(u))$. Since $\hat{C} \subseteq X \cup D(u)$, this means $\pi(\hat{C}) = \pi(X \cup D(u))$. Lemma 25 then implies $|\pi_K((X \cup D(u)) \setminus K)| \geq \hat{g}_K(X \cup D(u))$. We also have $|\pi_K(D(u) \setminus K)| = \hat{g}_K(D(u))$ by Lemma 26 and $|\pi_K((X \cap D(u)) \setminus K)| \geq \hat{g}_K(X \cap D(u))$ by the inductive assumption. Since $|\pi_K(\,\cdot \setminus K)| : 2^U \to \mathbf{Z}$ is submodular and $\hat{g}_K$ is intersecting supermodular, we obtain

$$\begin{aligned} |\pi_K(X \setminus K)| &\geq |\pi_K((X \cup D(u)) \setminus K)| + |\pi_K((X \cap D(u)) \setminus K)| - |\pi_K(D(u) \setminus K)| \\ &\geq \hat{g}_K(X \cup D(u)) + \hat{g}_K(X \cap D(u)) - \hat{g}_K(D(u)) \\ &\geq \hat{g}_K(X), \end{aligned}$$

which completes the proof. □

6 Extension to Skew-supermodular Coloring

This section extends Theorem 4 to the setting of skew-supermodular coloring. A function $g \colon 2^U \to \mathbf{Z} \cup \{-\infty\}$ is called *skew-supermodular* if every pair of $X, Y \subseteq U$ satisfies either the supermodular inequality or the *negamodular inequality* $g(X) + g(Y) \leq g(X \setminus Y) + g(Y \setminus X)$. By definition, skew-supermodularity is a generalization of intersecting supermodularity. It is known that Theorem 3 remains true for a pair of skew-supermodular functions [5].

For a skew-supermodular function, define its reduction by a subset $K \subseteq U$ as in Section 3. Then, we can confirm that it is again skew-supermodular, i.e., Claim 7 can extend to skew-supermodular functions. Also, proofs in Section 4 do not depend on intersecting supermodularity except that they use Claim 7

and the existence of skeleton posets. Therefore, to extend Theorem 4, it suffices to show the existence of skeleton posets for skew-supermodular functions. Let $g\colon 2^U \to \mathbf{Z} \cup \{-\infty\}$ be a skew-supermodular function and $\pi : U \to [k]$ be a minimal g-dominating k-coloring. Then, we can check that the following claims hold.

Claim 28 *If the conditions* (1)–(3) *hold with $F = X$ and $F = Y$, then X and Y satisfy the supermodular inequality of g.*

By Claim 28, we can extend Claim 17 for skew-supermodular functions. Therefore, we can define $D(u)$ for each $u \in U$ similarly to the case of intersecting-supermodular functions.

Claim 29 *For any $u \in U$ and any tight set $X \subseteq U$ with $D(u) \cap X \neq \emptyset$, $D(u)$ and X satisfy the supermodular inequality of g.*

Claim 29 says that, for $D(u)$ and tight set X with $D(u) \cap X \neq \emptyset$, the skew-supermodularity implies the supermodular inequality. Observe that, in the proofs after Claim 17, we apply the supermodular inequality only for such pairs of subsets. Thus, the same arguments work for skew-supermodular functions, and we obtain the following extension of Theorem 4.

Theorem 30 *For skew-supermodular functions $g_1, g_2 : 2^U \to \mathbf{Z} \cup \{-\infty\}$ and $k \in \mathbf{Z}_{>0}$, assume that there exists a k-coloring which dominates both g_1 and g_2. If L satisfies $|L(u)| = k$ for each $u \in U$, then there exists a coloring $\varphi : U \to \Sigma$ such that every $u \in U$ satisfies $\varphi(u) \in L(u)$ and φ dominates both g_1 and g_2.*

References

[1] M. Aigner and G. M. Ziegler: *Proofs from the Book*, Springer-Verlag, Berlin & Heidelberg, 2010.

[2] EGRES: Open problems: `http://lemon.cs.elte.hu/egres/open/` (accessed October 28, 2016).

[3] T. Fleiner: A fixed-point approach to stable matchings and some applications, *Mathematics of Operations Research*, **28** (2003), pp. 103–126.

[4] T. Fleiner and Z. Jankó: On weighted kernels of two posets, *Order*, **33** (2016), pp. 51–65.

[5] A. Frank and T. Király: A survey on covering supermodular functions, *Research Trends in Combinatorial Optimization* (W. J. Cook, L. Lovász, and J. Vygen, eds.), Springer-Verlag,, 2009, pp. 87–126.

[6] A. Frank, T. Király, J. Pap, and D. Pritchard: Characterizing and recognizing generalized polymatroids, *Mathematical Programming*, **146** (2014), pp. 245–273.

[7] D. Gale and L. S. Shapley: College admissions and the stability of marriage, *American Mathematical Monthly*, **69** (1962), pp. 9–15.

[8] F. Galvin: The list chromatic index of a bipartite multigraph, *Journal of Combinatorial Theory, Series B*, **63** (1995), pp. 153–158.

[9] D. Kőnig: Graphok és alkalmazásuk a determinánsok és a halmazok elméletére (Hungarian; Graphs and their application to the theory of determinants and sets), *Mathematikai és Természettudományi Értesitő*, **34** (1916), pp. 104–119.

[10] B. Sands, N. Sauer, and R. Woodrow: On monochromatic paths in edge-coloured digraphs, *Journal of Combinatorial Theory, Series B*, **33** (1982), pp. 271–275.

[11] A. Schrijver: Supermodular colourings, *Matroid Theory* (L. Lovász and A. Recski, eds.), North-Holland, Amsterdam, 1985, pp. 327–343.

[12] É. Tardos: Generalized matroids and supermodular colourings, *Matroid Theory* (L. Lovász and A. Recski, eds.), North-Holland, Amsterdam, 1985, pp. 359–382.

Global rigidity of generic frameworks on the cylinder

Bill Jackson

School of Mathematical Sciences
Queen Mary, University of London
London, UK
b.jackson@qmul.ac.uk

Anthony Nixon

Department of Mathematics and Statistics
Lancaster University
Lancaster, UK
a.nixon@lancaster.ac.uk

Abstract: We show that a generic framework (G, p) in $\mathbb{R}^3$ whose vertices are constrained to lie on the cylinder is globally rigid if and only if G is a complete graph on at most four vertices or G is both redundantly rigid and 2-connected. To prove the theorem we derive a new recursive construction of circuits in the simple $(2, 2)$-sparse matroid.

Keywords: rigidity, global rigidity, circuit, stress matrix, framework on a surface

1 Introduction

A (bar-joint) framework (G, p) in $\mathbb{R}^d$ is the combination of a finite, simple graph $G = (V, E)$ and a realisation $p : V \to \mathbb{R}^d$. The framework (G, p) is rigid if every edge-length preserving continuous motion of the vertices arises as a congruence of $\mathbb{R}^d$. Moreover (G, p) is globally rigid if every framework (G, q) with the same edge lengths as (G, p) arises from a congruence of $\mathbb{R}^d$.

In general it is NP-hard to determine the rigidity or global rigidity of a given framework [1, 14]. These problems become more tractable, however, for generic frameworks. It is known that both the rigidity and global rigidity of a generic framework (G, p) in $\mathbb{R}^d$ depend only on the underlying graph G, see [2, 4]. We say that G *is rigid* or *globally rigid in* $\mathbb{R}^d$ if some/every generic realisation of G in $\mathbb{R}^d$ has the corresponding property. Combinatorial characterisations of generic rigidity and global rigidity in $\mathbb{R}^d$ have been obtained when $d \leq 2$, see [10, 5], and these characterisations give rise to efficient combinatorial algorithms to decide if these properties hold. In higher dimensions, however, no combinatorial characterisations or algorithms are yet known.

We consider the situation where (G, p) is a framework in $\mathbb{R}^3$ whose vertices are constrained to lie on a fixed surface. Combinatorial characterisations for generic rigidity in this context were established for surfaces consisting of spheres [15, 12], cylinders [12], and cones [13]. In particular it was shown that a generic realisation of a graph G on a family of concentric spheres is rigid if and only G is rigid in the plane.

Connelly and Whiteley [3] showed that a graph G is generically globally rigid on the sphere if and only if it is generically globally rigid in the plane (which holds if and only if G is 3-connected and redundantly rigid in the plane by [5]). In [7], necessary combinatorial conditions were established for a framework on a surface to be generically globally rigid. The conditions, redundant rigidity and k-connectivity (where the integer k depends on the chosen surface), are analogous to those which characterise generic global rigidity on the plane and the sphere. These conditions were conjectured to also be sufficient for cylinders and cones. We verify this conjecture for the cylinder.

2 Rigidity and Stress Matrices

Let $G = (V, E)$ be a graph with $V = \{v_1, \ldots, v_n\}$. We will consider realisations of G on the unit cylinder $\mathcal{Y} = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 = 1\}$. (For the purposes of global rigidity there is no loss in generality

in assuming our cylinder has unit radius and is centred on the z-axis.) A *framework* (G,p) on $\mathcal{Y}$ is an ordered pair consisting of a graph G and a realisation p such that $p(v_i) \in \mathcal{Y}$ for all $v_i \in V$.

Two frameworks (G,p) and (G,q) on $\mathcal{Y}$ are *equivalent* if $\|p(v_i)-p(v_j)\| = \|q(v_i)-q(v_j)\|$ for all edges $v_iv_j \in E$. Moreover (G,p) and (G,q) are *congruent* if $\|p(v_i)-p(v_j)\| = \|q(v_i)-q(v_j)\|$ for all pairs of vertices $v_i, v_j \in V$. The framework (G,p) is *globally rigid* on $\mathcal{Y}$ if every equivalent framework (G,q) on $\mathcal{Y}$ is congruent to (G,p). It is *rigid* on $\mathcal{Y}$ if there exists an $\epsilon > 0$ such that every framework (G,q) on $\mathcal{Y}$ which is equivalent to (G,p), and has $\|p(v_i)-q(v_i)\| < \epsilon$ for all $1 \le i \le n$, is congruent to (G,p). It is *redundantly rigid* on $\mathcal{Y}$ if $(G-e,p)$ is rigid on $\mathcal{Y}$ for all $e \in E$. It is *generic* on $\mathcal{Y}$ if $\operatorname{td}[\mathbb{Q}(p):\mathbb{Q}] = 2n$.

An *infinitesimal flex* s of (G,p) on $\mathcal{Y}$ is a map $s : V \to \mathbb{R}^3$ such that $s(v_i)$ is tangential to $\mathcal{Y}$ at $p(v_i)$ for all $v_i \in V$ and $(p(v_j)-p(v_i))\cdot(s(v_j)-s(v_i)) = 0$ for all $v_jv_i \in E$. The framework (G,p) is *infinitesimally rigid* on $\mathcal{Y}$ if every infinitesimal flex is an infinitesimal isometry of $\mathbb{R}^3$. It was shown in [12] that a generic framework (G,p) on $\mathcal{Y}$ is rigid if and only if it is a complete graph on at most three vertices or is infinitesimally rigid.

The *rigidity matrix* $R_{\mathrm{cyl}}(G,p)$ is the $(|E|+|V|)\times 3|V|$ matrix

$$R_{\mathrm{cyl}}(G,p) = \begin{pmatrix} R(G,p) \\ S(G,p) \end{pmatrix}$$

where: $R(G,p)$ has rows indexed by E and 3-tuples of columns indexed by V in which, for $e = v_iv_j \in E$, the submatrices in row e and columns v_i and v_j are $p(v_i)-p(v_j)$ and $p(v_j)-p(v_i)$, respectively, and all other entries are zero; $S(G,p)$ has rows indexed by V and 3-tuples of columns indexed by V in which, for $v_i \in V$, the submatrix in row v_i and column v_i is $\bar{p}(v_i) = (x_i, y_i, 0)$ when $p(v_i) = (x_i, y_i, z_i)$.

An *equilibrium stress* for a framework (G,p) on $\mathcal{Y}$ is a pair (ω,λ), where $\omega : E \to \mathbb{R}$ and $\lambda : V \to \mathbb{R}$ and (ω,λ) belongs to the cokernel of $R_{\mathrm{cyl}}(G,p)$. Thus (ω,λ) is an equilibrium stress for (G,p) on $\mathcal{Y}$ if and only if

$$\sum_{j=1}^{n} \omega_{ij}(p(v_i)-p(v_j)) + \lambda_i\bar{p}(v_i) = 0 \text{ for all } 1 \le i \le n, \tag{1}$$

where ω_{ij} is taken to be equal to ω_e if $e = v_iv_j \in E$ and to be equal to 0 if $v_iv_j \notin E$.

Given a stress (ω,λ) for a framework (G,p) on $\mathcal{Y}$ we define $\Omega = \Omega(\omega)$ to be the $n\times n$ symmetric matrix with off-diagonal entries $-\omega_{ij}$ and diagonal entries $\sum_j \omega_{ij}$, and $\Lambda = \Lambda(\lambda)$ to be the $n\times n$ diagonal matrix with diagonal entries $\lambda_1, \lambda_2, \ldots, \lambda_n$. The *stress matrix* associated to (ω,λ) is the $3n \times 3n$ symmetric matrix

$$\Omega_{\mathrm{cyl}}(\omega,\lambda) = \begin{bmatrix} \Omega+\Lambda & 0 & 0 \\ 0 & \Omega+\Lambda & 0 \\ 0 & 0 & \Omega \end{bmatrix}.$$

It follows from the definitions of Ω, Λ and (1) that $(x_1, x_2, \ldots, x_n), (y_1, y_2, \ldots, y_n)$ are in the cokernel of $\Omega+\Lambda$ and $(z_1, z_2, \ldots, z_n), (1, 1, \ldots, 1)$ are in the cokernel of Ω. This implies that $\operatorname{rank}\Omega_{\mathrm{cyl}}(\omega,\lambda) \le 3n-6$. We say that (ω,λ) *has maximum rank* when $\operatorname{rank}\Omega_{\mathrm{cyl}}(\omega,\lambda) = 3n-6$.

3 Main result

Our main result is the following theorem giving a combinatorial characterisation of global rigidity. Note also that the combinatorial conditions can be checked efficiently.

Theorem 1 *Let (G,p) be a generic framework on $\mathcal{Y}$. Then (G,p) is globally rigid on $\mathcal{Y}$ if and only if G is either a complete graph on at most four vertices or G is 2-connected and redundantly rigid on the cylinder.*

The necessary conditions were proved in [7]. The key part in proving sufficiency in Theorem 1 is to deal with the special case when G is 2-connected and redundantly rigid with the minimum possible number of edges. It follows from [12] that such graphs can be defined purely combinatorially as follows.

The *simple* $(2,2)$*-sparse matroid* for a graph G is the matroid $\mathcal{M}^*_{2,2}(G)$ on $E(G)$ in which a set of edges F is *independent* if and only if $|F'| \leq 2|V(F')| - 2$ for all $\emptyset \neq F' \subseteq F$, with strict inequality when $|F'| = 2$. We will abuse terminology and refer to the graphs induced by circuits in $\mathcal{M}^*_{2,2}$ as $\mathcal{M}^*_{2,2}$*-circuits*.

To show every $\mathcal{M}^*_{2,2}$-circuit is globally rigid we use the constructive characterisation given in the following section. We then show that each of the operations in our constructions preserve the property of having a maximum rank equilibrium stress using 'special position' arguments and a delete-contract characterisation of infinitesimal rigidity for frameworks on the cylinder with two coincident points from [6]. We can then use a result from [8] to show that any two equivalent generic frameworks (G,p) and (G,p') on the cylinder have closely related maximum rank equilibrium stresses. This tells us that $(z'_1, z'_2, \ldots, z'_n)$ is a scalar multiple of $(z_1, z_2, \ldots, z_n)$. We complete the proof that these two frameworks are congruent by using a characterisation of global rigidity for generic frameworks on the cylinder with the added constraint that the projection of a motion onto the axis of the cylinder is a dilation. Full details can be found in [9].

4 Recursive construction of $\mathcal{M}^*_{2,2}$-circuits

Given an $\mathcal{M}^*_{2,2}$-circuit $G = (V,E)$, the first operation in our construction, K_4^-*-extension*, deletes an edge v_1v_2 and adds two new vertices u_1, u_2 along with 5 new edges $u_1u_2, u_1v_1, u_1v_2, u_2v_1, u_2v_2$. The second operation, *generalised vertex split*, is defined as follows: choose $v \in V$ and a partition N_1, N_2 of the neighbours of v; then delete v from G and add two new vertices v_1, v_2 joined to N_1, N_2, respectively; finally add two new edges v_1v_2, v_1x for some $x \in V \setminus N_2$. These operations are illustrated in Figures 1 and 2, and the base graphs for our construction are shown in Figure 3(a) and (b).

Figure 1: K_4^--extension.

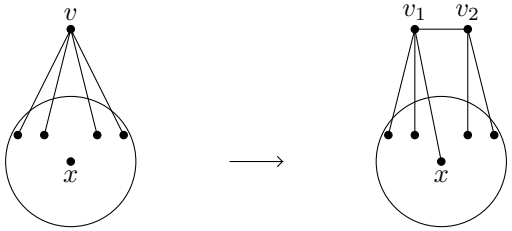


Figure 2: Generalised vertex split.

Theorem 2 *Suppose G is an $\mathcal{M}^*_{2,2}$-circuit. Then G can be obtained from either $K_5 - e$ or H_1 by recursively applying the operations of K_4^--extension and generalised vertex split, in such a way that each of the intermediate graphs is an $\mathcal{M}^*_{2,2}$-circuit.*

To prove Theorem 2 we use two results from [11]. The first is a decomposition result for $\mathcal{M}^*_{2,2}$-circuits which uses the graph operations defined in Figure 4.

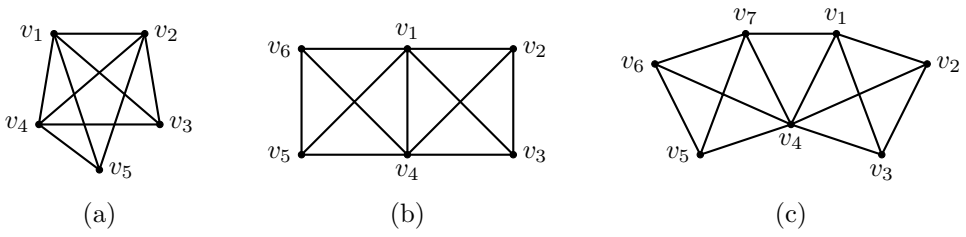


Figure 3: The graphs $K_5 - e, H_1$ and H_2.

Theorem 3 *[11, Lemmas 3.1, 3.2, 3.3] Suppose G_0, G_1, G_2 are graphs with $|E(G_i)| = 2|V(G_i)| - 2$ for all $0 \leq i \leq 2$ and that G_0 is an j-join of G_1 and G_2 for some $1 \leq j \leq 3$. Then G_0 is a $\mathcal{M}^*_{2,2}$-circuit if and only if both G_1 and G_2 are $\mathcal{M}^*_{2,2}$-circuits.*

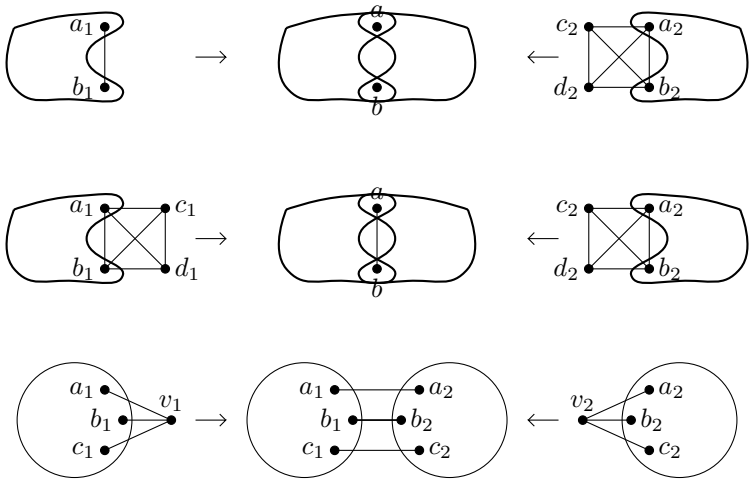


Figure 4: The 1-, 2- and 3-join operations. The 1- and 2-join operations form the graphs in the centre by merging a_1 and a_2 into a, and b_1 and b_2 into b.

The second result we use is a recursive construction for $\mathcal{M}^*_{2,2}$-circuits which uses the i-join operations as well as the 1-*extension operation* which deletes an edge xy from a graph G and then adds a new vertex v and three new edges vx, vy, vz for some vertex $z \neq x, y$. The recursion begins with the three $\mathcal{M}^*_{2,2}$-circuits defined in Figure 3.

Theorem 4 *[11, Theorem 1.1] Suppose G is an $\mathcal{M}^*_{2,2}$-circuit. Then G can be obtained from either $K_5 - e$, H_1 or H_2 by recursively applying the operations of* 1*-extension, and* 1*-,* 2*- and* 3*-join.*

Now consider generalised vertex splitting. The usual vertex splitting operation, see [16], is the special case when x is chosen to be a neighbour of v_2. Note also that the special case when v_1 has degree 3 (and $v_2 = v$) is the 1-extension operation. This observation allows us to use Theorem 4 to reduce the proof of Theorem 2 to considering small cutsets. To do this we use Theorem 3 and a case by case analysis.

The following construction shows the K_4^--extension operation is required in Theorem 2. Take any $\mathcal{M}^*_{2,2}$-circuit H, and apply the K_4^--extension to every single edge of H. The resulting graph G has two types of edges. Those edges with no end-vertices in H are contained in two triangles so any K_4^--reduction which contracts such an edge results in a non-simple graph. The remaining edges, those with exactly one

end-vertex in H, are not admissible either since any K_4^--reduction which contracts such an edge results in a graph containing either a multiple edge or a vertex of degree two.

References

[1] T. Abbott, Generalizations of Kempe's universality theorem, *Master's thesis, Massachusetts Institute of Technology* (2008).

[2] L. Asimow and B. Roth, The rigidity of graphs, *Trans. Amer. Math. Soc.* **245** (1978), 279–289.

[3] R. Connelly and W. Whiteley, Global rigidity: the effect of coning, *Discrete Comput. Geom.* **43**:4 (2010) 717-735.

[4] S. Gortler, A. Healy and D. Thurston, Characterizing generic global rigidity, *Amer. J. Math.* **132**:4 (2010) 897-939.

[5] B. Jackson and T. Jordán, Connected Rigidity Matroids and Unique Realisations of Graphs, *J. Comb. Theory B* **94** (2005) 1-29.

[6] B. Jackson, V. Kaszanitzky and A. Nixon, Rigid cylindrical frameworks with two coincident points, *arxiv: 1607:02039*, 2016

[7] B. Jackson, T. McCourt and A. Nixon, Necessary conditions for the generic global rigidity of frameworks on surfaces, *Discrete and Comput. Geom.*, **52**:2 (2014) 344-360.

[8] B. Jackson and A. Nixon, Stress matrices and global rigidity of frameworks on surfaces, *Discrete and Comput. Geom.*, **54**:3 (2015) 586-609.

[9] B. Jackson and A. Nixon, Global rigidity of generic frameworks on the cylinder, *arXiv: 1610.07755*

[10] G. Laman, On graphs and rigidity of plane skeletal structures, *J. Engineering Math.* **4** (1970), 331-340.

[11] A. Nixon, A constructive characterisation of circuits in the simple $(2,2)$-sparse matroid, *European Journal of Combinatorics* **42** (2014), 92–106.

[12] A. Nixon, J. Owen and S. Power, Rigidity of frameworks supported on surfaces, *SIAM Journal on Discrete Mathematics* **26**:4 (2012), 1733–1757.

[13] A. Nixon, J. Owen and S. Power, A characterization of generically rigid frameworks on surfaces of revolution, *SIAM Journal on Discrete Mathematics* **28**:4 (2014), 2008–2028.

[14] J. Saxe, Embeddability of weighted graphs in k-space is strongly NP-hard, *In Seventeenth Annual Allerton Conference on Communication, Control, and Computing, Proceedings of the Conference held in Monticello, Ill., October 10-12, 1979.*

[15] W. Whiteley, Cones, infinity and one story buildings, *Structural Topology*, **8** (1983), 53–70.

[16] W. Whiteley, Vertex splitting in isostatic frameworks, *Structural Topology*, 16 (1990) 23–30.

Equivalent Realisations of Rigid Graphs

Bill Jackson

School of Mathematical Sciences
Queen Mary University of London
Mile End Road, London E1 4NS, UK
b.jackson@qmul.ac.uk

J.C. Owen

Siemens
Park House, Cambridge CB3 0DU, UK
owen.john.ext@siemens.com

Abstract: Given a rigid framework (G, p) in $\mathbb{R}^2$, we consider the problem of determining the maximum number of pairwise non-congruent rigid frameworks (G, q) which have the same edge lengths as (G, p). This problem can be restated as finding the number of solutions of a related system of quadratic equations and in this context it is natural to consider the complex solutions to obtain a better understanding of the real solutions. We show that the number of complex solutions, $\text{comp}(G)$, is the same for all generic realisations (G, p), characterise the graphs G for which $\text{comp}(G) = 1$, and show that the problem of determining $\text{comp}(G)$ can be reduced to the case when G is 3-connected and has no non-trivial 3-edge-cuts. We also consider the effect of the Henneberg moves and the vertex-splitting operation on $\text{comp}(G)$.

Keywords: rigid frameworks, globally rigid frameworks, equivalent realisations

1 Introduction

Graphs with geometrical constraints provide natural models for a variety of applications, including Computer-Aided Design, sensor networks and flexibility in molecules. Given a straight line realisation of a graph in Euclidean d-dimensional space $\mathbb{R}^d$, a fundamental problem is to determine whether this realisation is unique or, more generally, determine how many distinct realisations exist with the same edge lengths. Saxe [15] showed that the uniqueness problems is NP-hard, but this hardness relies on algebraic relations between coordinates of vertices and for practical purposes it is natural to study generic realisations. Gortler, Healy and Thurston [8] showed that the uniqueness of a generic realisation in $\mathbb{R}^d$ depends only on the structure of the underlying graph, and graphs with the property that all their generic realisations in $\mathbb{R}^d$ are unique have been characterised when $d = 1, 2$, see [11].

In contrast, the number of realisations which are equivalent to, i.e. have the same edge lengths as, a given generic realisation of a graph in $\mathbb{R}^d$ may depend on both the graph and the realisation when $d \geq 2$, see Figures 1 and 2. Bounds on the maximum number of equivalent realisations in $\mathbb{R}^2$, where the maximum is taken over all possible realisations of a given graph, were obtained by Borcea and Streinu in [2], and this number is determined exactly for generic realisations of graphs with a connected rigidity matroid by Jackson, Jordán, and Szabadka in [12].

The set of all realisations which are equivalent to a given realisation can be represented as the set of solutions to a system of quadratic equations. In this setting it is natural to consider the number of complex solutions. This number gives an upper bound on the number of real solutions which often plays a crucial role in calculating the exact number of real solutions, see for example [5, 4, 16]. In addition, the number of complex solutions is much better behaved than the number of real solutions. It is known, for example, that the number of complex solutions is the same for all generic realisation of given graph, see [13]. The realisations of the graph G shown in Figures 1 and 2 both have four equivalent complex realisations. Only two of these are real in Figure 1, but all four are real in Figure 2.

Definitions and a preliminary result

A *d-dimensional complex*, respectively *real, framework* (G,p) is a graph $G=(V,E)$ together with a map p from V to $\mathbb{C}^d$, respectively $\mathbb{R}^d$. We will also refer to the ordered pair (G,p) as a *realisation of* G *in* $\mathbb{C}^d$ *or* $\mathbb{R}^d$. A framework (G,p) is *generic* if the set of all coordinates of the points $p(v)$, $v\in V$, is algebraically independent over $\mathbb{Q}$. We will restrict our attention to 2-dimensional frameworks unless explicitly stated otherwise.

For $P=(x,y)\in\mathbb{C}^2$ let $d(P)=x^2+y^2$. and $\|P\|=|x|^2+|y|^2$, where $|.|$ denotes the modulus of a complex number. Two frameworks (G,p) and (G,q) are *equivalent* if $d(p(u)-p(v))=d(q(u)-q(v))$ for all $uv\in E$, and are *congruent* if $d(p(u)-p(v))=d(q(u)-q(v))$ for all $u,v\in V$.

A framework (G,p) is *complex*, respectively *real*, *globally rigid* if every complex, respectively real, framework (G,q) which is equivalent to (G,p), is congruent to (G,p). It is *complex*, respectively *real*, *rigid* if there exists an $\epsilon>0$ such that every complex, respectively real, framework (G,q) which is equivalent to (G,p) and satisfies $\|(p(v)-q(v)\|<\epsilon$ for all $v\in V$, is congruent to (G,p). It is known that real/complex rigidity and real/complex global rigidity are generic properties of frameworks and that a graph is generically real (globally) rigid if and only if it is generically complex (globally) rigid, see [13, 9]. Hence we may describe a graph as being *rigid* or *globally rigid* if every generic realisation in $\mathbb{R}^2$ (or equivalently $\mathbb{C}^2$) has these properties. Graphs which are generically rigid or globally rigid (in $\mathbb{R}^2$ or $\mathbb{C}^2$) are characterised in [14] and [11], respectively.

Given a rigid complex, respectively real, framework (G,p), we let $\mathrm{comp}(G,p)$, respectively $\mathrm{real}(G,p)$, denote the number of congruence classes in the set of all complex, respectively real, rigid frameworks which are equivalent to (G,p). We show in [13] that both these numbers are finite, and that $\mathrm{comp}(G,p)$ is the same for all generic realisations of G in $\mathbb{C}^2$. We put $\mathrm{comp}(G)=\mathrm{comp}(G,p)$ for any generic realisation (G,p). The following result shows that $\mathrm{comp}(G,p)$ gives a lower bound on $\mathrm{comp}(G)$ when (G,p) is a rigid (but not necessarily generic) realisation of a *minimally rigid* graph, i.e. G is rigid but $G-e$ is not rigid for all edges e of G.

Theorem 1 *[13] Let (G,p) be a rigid realisation of a minimally rigid graph G. Then* $\mathrm{comp}(G)\geq \mathrm{comp}(G,p)+\widehat{\mathrm{comp}}(G,p)$*, where $\widehat{\mathrm{comp}}(G)$ is the number of congruence classes in the set of realisations which are equivalent to (G,p) and are rigid, but not infinitesimally rigid.*

The intuition behind Theorem 1 is as follows. Suppose we choose a sequence of generic frameworks (G,p^k) with $p^k\to p$. This gives rise to $\mathrm{comp}(G)$ sequences of frameworks (G,p_i^k) with (G,p_i^k) equivalent to (G,p^k). Each rigid framework (G,p_j) which is equivalent to (G,p) is a limit point of at least one of the sequences (G,p_i^k), with equality only if (G,p_j) is infinitesimally rigid. This gives $\mathrm{comp}(G)\geq \mathrm{comp}(G,p)+\widehat{\mathrm{comp}}(G,p)$.

Theorem 1 tells us that we can calculate $\mathrm{comp}(G)$ when G is minimally rigid by first determining an upper bound, say $\mathrm{comp}(G)\leq k$, and then exhibiting a particular realisation (G,p) with the property that $\mathrm{comp}(G,p)+\widehat{\mathrm{comp}}(G,p)\geq k$. A worked example is given in Section 4.

2 Main results

We first consider the effect on $\mathrm{comp}(G)$ of three operations which are commonly used to give recursive constructions for families of rigid graphs. Given a graph $H=(V,E)$ with $u\in V$ and $e\in E$, the 0-*extension operation* adds a new vertex v and two new edges from v to H. The 1-*extension operation* deletes e and then adds a new vertex v and three new edges from v to $H-e$, two of which go to the end-vertices of e. The *vertex split operation* deletes u and then adds two new vertices u_1,u_2 and edges $u_1u_2,u_1x_1,u_1x_2,\ldots,u_1x_k$ and $u_2x_1,u_2x_{k+1},\ldots,u_2x_t$ where $\{x_1,x_2,\ldots,x_t\}$ is the neighbour set of u in H. It is known that each of these operations preserve the properties of being rigid or minimally rigid.

Theorem 2 *Suppose that H is a rigid graph.*
(a) If G is obtained from H by a 0-extension operation then $\mathrm{comp}(G)=2\,\mathrm{comp}(H)$*. Furthermore, if (H,p) is a generic real realisation of H, then there exists a generic real realisation (G,q) of G such that*

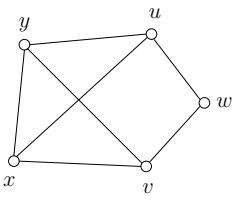


Figure 1: A framework in $\mathbb{R}^2$. The only equivalent but non-congruent framework can be obtained by reflecting the vertex w in the line through $\{u, v\}$, giving a total of two real equivalent but non-congruent realisations.

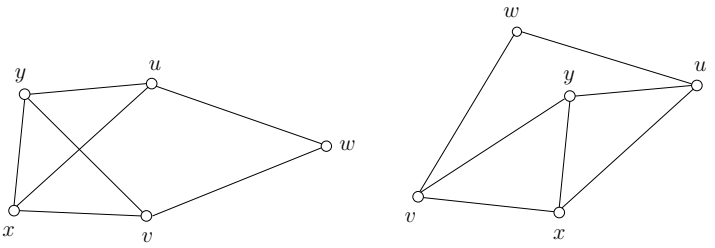


Figure 2: Two equivalent but non-congruent realisations of the graph G of Figure 1 in $\mathbb{R}^2$. Two other equivalent but non-congruent realisation can be obtained from these by reflecting the vertex w in the line through $\{u, v\}$, giving a total of four equivalent but non-congruent realisations in $\mathbb{R}^2$.

$\text{real}(G, q) = 2\,\text{real}(H, p)$.
(b) If G is obtained from H by a 1-extension operation which deletes the edge e and $H - e$ is rigid then $\text{comp}(G) = \text{comp}(H)$. *Furthermore, if (G, p) is a generic real realisation of G, then* $\text{real}(G, p) = \text{real}(H, p|_H)$.
(c) If H is minimally rigid and G is obtained from H by a vertex splitting operation then $\text{comp}(G) \geq 2\,\text{comp}(H)$. *Furthermore, if (H, p) is a generic real realisation of H, then there exists a generic real realisation (G, q) of G such that* $\text{real}(G, q) \geq 2\,\text{real}(H, p)$.

The proof of (a) is straightforward, see for example [2]. The proofs of (b) and (c) are given in [13]. For (b) we use similar ideas to those developed in [12] to show that the 1-extension operation preserves the global rigidity of redundantly rigid graphs. More precisely, if v is the vertex added to H by the 1-extension operation, we use the rigidity of $H - e = G - v$ and the algebraic independence of the coordinates of a generic realisation (G, p) to show that the neighbours of v are *globally linked* in (G, p) i.e. the distance between them is the same in every equivalent realisation. For (c), we choose a generic framework (H, p) and then consider the special realisation (G, q) which we obtain from (H, p) by putting both of the new vertices in the same position as the deleted vertex. Then use Theorem 1 and the fact that all equivalent realisations to (G, q) are rigid but not infinitesimally rigid.

Theorem 2(b) is used in [13] to determine $\text{comp}(G)$ when G has a connected rigidity matroid. In particular, we show that a graph G has $\text{comp}(G) = 1$ if and only if it is a complete graph on at most three vertices or is 2-connected and redundantly rigid. This is the same characterization as that given for graphs which are generically globally rigid in $\mathbb{R}^2$ in [11] and allows us to deduce that generic global rigidity in $\mathbb{R}^2$ and $\mathbb{C}^2$ are equivalent. Gortler and Thurston [9] use stress matrices to show that this equivalence extends to $\mathbb{R}^d$ and $\mathbb{C}^d$.

We next consider graphs which have small separating sets.

Theorem 3 *[13] Suppose that G_1, G_2 are subgraphs of a rigid graph G and that $G = G_1 \cup G_2$.*
(a) If $G_1 \cap G_2$ is a globally rigid graph on at least three vertices then G_1, G_2 are both rigid and

$\mathrm{comp}(G) = \mathrm{comp}(G_1)\,\mathrm{comp}(G_2)$. *Furthermore, if* (G,p) *is a generic real realisation of* G*, then* $\mathrm{real}(G,p) = \mathrm{real}(G,p|_{G_1})\mathrm{real}(G,p|_{G_2})$.
(b) If $V(G_1)\cap V(G_2) = \{u,v\}$ *and* G_1, G_2 *are both rigid then* $\mathrm{comp}(G) = 2\,\mathrm{comp}(G_1+uv)\,\mathrm{comp}(G_2+uv)$. *Furthermore, if* (G,p) *is a generic real realisation of* G*, then* $\mathrm{real}(G,p) = 2\,\mathrm{real}(G,p|_{G_1})\mathrm{real}(G,p|_{G_2})$.
(c) If $V(G_1)\cap V(G_2) = \{u,v\}$ *and* G_2 *is not rigid then* G_1, G_2+uv *are both rigid and* $\mathrm{comp}(G) = 2\,\mathrm{comp}(G_1)\,\mathrm{comp}(G_2+uv)$.

The proof of (a) is straightforward. The proofs of (b) and (c) are given in [13]. For (b) we again use ideas from [12] to show that the vertices u, v are globally linked in (G,p). For (c), we use the fact that $\mathbb{C}$ is algebraically closed to show that every equivalent framework to $(G_1, p|_{G_1})$ can be extended to $\mathrm{comp}(G_2+uv)$ frameworks which are equivalent to (G,p).

Theorem 4 *[13] Suppose that* G_1, G_2 *are disjoint subgraphs of a rigid graph* G *and that* $G = G_1 \cup G_2 \cup \{e_1, e_2, e_3\}$ *for three independent edges* e_1, e_2, e_3 *of* G*. Then* G_1, G_2 *are both rigid and* $\mathrm{comp}(G) = 12\,\mathrm{comp}(G_1)\,\mathrm{comp}(G_2)$.

The proof Theorem 4 first uses the the fact that $\mathbb{C}$ is algebraically closed to show that $\mathrm{comp}(G) = \mathrm{comp}(G_1)\,\mathrm{comp}(G_2^*)$, where G_2^* is obtained from G by replacing G_1 by a triangle. We may deduce similarly that $\mathrm{comp}(G_2^*) = \mathrm{comp}(G_2)\,\mathrm{comp}(P)$ where P is the triangular prism. We complete the proof by using the fact that $\mathrm{comp}(P) = 12$.

Theorems 3 and 4 can be used to reduce the problem of determining $\mathrm{comp}(G)$ to the case when G is 3-connected and has no nontrivial 3-edge-cuts.

3 Examples and Open Problems

The obvious open problem is:

Problem 5 *Can* $\mathrm{comp}(G)$ *be determined efficiently for an arbitrary rigid graph* G*?*

The minimally rigid graphs G_1, G_2, G_3 and G_4 of Figure 3 indicate that it may be difficult to obtain an affirmative answer to Problem 5 for all graphs. Emeris and Moroz [4] give a real framework (G_1,p) with $\mathrm{real}(G,p) = 28$ and use mixed volume techniques to prove that $\mathrm{comp}(G_1,q) \leq 28$ for all complex rigid frameworks (G_1,q), (an error in their proof was subsequently corrected in [6]). We may now use Theorem 1 to deduce that $\mathrm{comp}(G_1) = 28$. A similar proof technique will be used show that $\mathrm{comp}(G_3) = 68$ in Section 4. Computer calculations, i.e. calculating $\mathrm{comp}(G,p)$ for 'randomly chosen' realisations (G,p), indicate that $\mathrm{comp}(G_2) = 22$ and $\mathrm{comp}(G_4) = 45$. These values have recently been confirmed by Josef Schicho (personal communication) using the algorithm described by Capco et al in [3]. It is difficult to imagine how these numbers could be deduced from the structures of G_1, G_2, G_3 and G_4.

Until recently, the fastest algorithms for determining $\mathrm{comp}(G)$ solved the associated system of polynomial equations using Gröbner basis calculations. Such algorithms are exponential and struggle to cope with some graphs on only seven vertices such as G_2. An exciting new algorithm based on a recurrence formula for $\mathrm{comp}(G)$ is described in [3]. Although still exponential, it has been used to determine $\mathrm{comp}(G)$ for all minimally rigid graphs on at most twelve vertices, see [3].

If we cannot determine $\mathrm{comp}(G)$ precisely then we could ask for tight asymptotic upper bounds on $\mathrm{comp}(G)$.

Problem 6 *Determine the smallest* $k \in \mathbb{R}$ *such that* $\mathrm{comp}(G) = O(k^n)$ *for all rigid graphs* G *with* n *vertices.*

Clearly $\mathrm{comp}(G)$ will be maximised when G is minimally rigid, and hence it follows from [2, Theorem 1.1] that $\mathrm{comp}(G) \leq \frac{1}{2}\binom{2n-4}{n-2} \approx 4^n$ for all rigid graphs G with n vertices. Borcea and Streinu [2, Proposition 5.6] also construct an infinite family of minimally rigid graphs G with $\mathrm{comp}(G) = 12^{(n-3)/3} \approx 2.29^n$ by taking several copies of the triangular prism P with a single triangle in common. The fact that

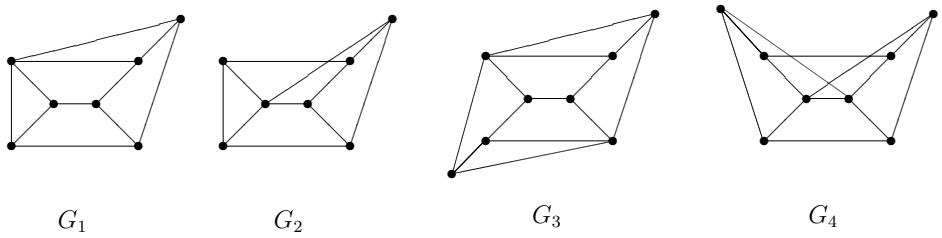


Figure 3: The graphs G_1, G_2, G_3 and G_4.

$\mathrm{comp}(G) = 12^{(n-3)/3}$ for this family can be deduced from Theorem 3(a) and the fact that $\mathrm{comp}(P) = 12$. Emiris and Moroz [4] use a similar construction with P replaced by G_1 to obtain an infinite family of minimally rigid graphs G with $\mathrm{comp}(G) = 28^{(n-3)/4} \approx 2.3^n$. A similar construction based on G_3 gives an infinite family of minimally rigid graphs G with $\mathrm{comp}(G) = 68^{(n-3)/5} \approx 2.33^n$.

The calculations in [3] determine the minimally rigid graphs on n vertices which maximise $\mathrm{comp}(G)$ for all $n \leq 12$. The graphs for $n = 6, 7, 8$ are the triangular prism P, G_1 and G_3. All three are planar graphs with exactly two triangles. The graphs from [3] for $n = 9, 10, 11$ are also planar with exactly two triangles and the corresponding values for $\mathrm{comp}(G)$ are 172, 440 and 1144, respectively. We may glue copies of their graph on 11 vertices together along a common triangle to obtain an infinite family of graphs with $\mathrm{comp}(G) = 1144^{(n-3)/8} \approx 2.41^n$. It follows that the answer to Problem 6 will satisfy $1144^{1/8} \leq k \leq 4$. (Curiously, their minimally rigid graph on $n = 12$ vertices which maximises $\mathrm{comp}(G)$ has no triangles so does not allow us to use this construction.)

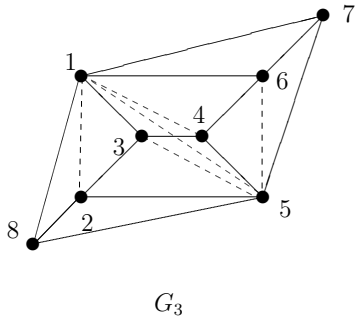
It would also be of interest to determine a tight lower bound on $\mathrm{comp}(G)$ when G is minimally rigid.

Conjecture 7 *For all minimally rigid graphs G with n vertices,* $\mathrm{comp}(G) \geq 2^{n-3}$.

The family of graphs which can be constructed from K_3 by 0-extension operations and Theorem 2(a) show that Conjecture 7 would be tight. We can show that this conjecture holds for planar graphs by using Theorem 2(c) and the recursive construction for minimally rigid planar graphs given in [7]. (Indeed we can obtain the stronger result that such a graph has at least 2^{n-3} pairwise equivalent generic *real* realisations.) The results of [3] confirm that the conjecture is also true when $n \leq 12$. Rather embarrassingly, the only lower bound we have for an arbitrary minimally rigid graph G is the trivial bound $\mathrm{comp}(G) \geq 2$.

Since every minimally rigid graph can be obtained from a triangle by 0- and 1-extensions, and since every 0-extension doubles $\mathrm{comp}(G)$ by Theorem 2(a), it is tempting to try to prove Conjecture 7 by showing that if we perform the 1-extension operation on a minimally rigid graph G then we will increase $\mathrm{comp}(G)$ by at least a factor of two. Unfortunately this is not the case: the graph G_2 of Figure 3 can be obtained from the triangular prism P by a 1-extension operation; we have $\mathrm{comp}(P) = 12$ and we have $\mathrm{comp}(G_2) = 22 < 2\,\mathrm{comp}(P) = 24$.

Dylan Thurston asked at a workshop on global rigidity held at Cornell University in February 2011 whether every rigid graph G has a generic real realisation (G, p) such that $\mathrm{real}(G, p) = \mathrm{comp}(G)$. The graph G_4 in Figure 3 shows that the answer to this question is negative: the proof technique used by Hendrickson [10] to obtain necessary conditions for global rigidity can be adapted to show that $\mathrm{real}(G, p)$ is even for all generic real realisations (G, p) of a graph G which is rigid but not globally rigid; on the other hand, we have $\mathrm{comp}(G_4) = 45$ which is odd. By glueing several copies of G_4 along a common edge and applying Theorem 3(b), we may construct an infinite family of graphs G on n vertices such that $\mathrm{real}(G)/\mathrm{comp}(G) \leq (44/45)^{\frac{n-2}{6}}$. It follows that we can make the ratio $\mathrm{real}(G)/\mathrm{comp}(G)$ arbitrarily close to zero. It would be of interest to find special families of graphs for which the answer to Thurston's question is positive.



G_3

Figure 4: The graph G_3.

4 A worked example

We will sketch a proof that $\text{comp}(G_3) = 68$ where the graph G_3 is as shown in Figure 3.

We first show that $\text{comp}(G_3) \leq 68$ using a similar technique to that described in [6]. Consider a generic realisation (G_4, p) of G in $\mathbb{C}^2$. The Cayley-Menger matrix CM for an arbitrary framework (G_3, q) which is equivalent to (G, p) is given by

$$CM = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & x_{12} & c_{13} & x_{14} & x_{15} & c_{16} & c_{17} & c_{18} \\ 1 & x_{12} & 0 & c_{23} & x_{24} & c_{25} & x_{26} & x_{27} & c_{28} \\ 1 & c_{13} & c_{23} & 0 & c_{34} & x_{35} & x_{36} & c_{37} & x_{38} \\ 1 & x_{14} & x_{24} & c_{34} & 0 & c_{45} & c_{46} & x_{47} & x_{48} \\ 1 & x_{15} & c_{25} & x_{35} & c_{45} & 0 & x_{56} & c_{57} & c_{58} \\ 1 & c_{16} & x_{26} & x_{36} & c_{46} & x_{56} & 0 & c_{67} & x_{68} \\ 1 & c_{17} & x_{27} & c_{37} & x_{47} & c_{57} & c_{67} & 0 & x_{78} \\ 1 & c_{18} & c_{28} & x_{38} & x_{48} & c_{58} & x_{68} & x_{78} & 0 \end{bmatrix}$$

where the first row and column are indexed as 0, the other rows and columns are indexed by the vertex labels as shown in Figure 4, $c_{ij} = d(p(i), p(j))$ for all edges ij of G_3, and the x_{ij} are indeterminates representing the values of $d(q(i), q(j))$ when $ij \notin E(G_3)$. The Cayley-Menger Theorem tells us that, since (G, q) is a framework in $\mathbb{C}^2$, every 5×5 principal minor of CM which contains the first row and column is equal to zero. In particular this gives us the following system of five polynomial equations involving the indeterminates $x_{12}, x_{14}, x_{15}, x_{35}, x_{56}$ shown by dashed lines in Figure 4:

$$CM(0,1,2,3,5) = CM(0,1,3,4,5) = CM(0,1,4,5,6) = CM(0,1,5,6,7) = CM(0,1,2,5,8) = 0. \quad (1)$$

Bernstein's Theorem [1] tells us that the number of solutions to this system of equations in $(\mathbb{C} \setminus \{0\})^5$ is bounded above by the *mixed volume* of the five Newton polytopes corresponding to the five polynomials in (1). This mixed volume is 68. Since G is rigid and (G, p) is generic, no equivalent framework (G, q) can have $d(q(i) - q(j)) = 0$ for any distinct i, j. Hence no solution to (1) can have a 0-component. Since the graph G_3^+ obtained by adding the dashed edges to G_3 is globally rigid, the realisation (G, q) is completely determined by the 'lengths' of the dashed edges i.e the values of $x_{12}, x_{14}, x_{15}, x_{35}, x_{56}$. Hence $\text{comp}(G_3) \leq 68$.

We next show that $\text{comp}(G_3) \geq 68$. It will suffice to exhibit a particular rigid realisation (G_3, p) with $\text{comp}(G_3, p) \geq 68$. Since G_3 is minimally rigid, we can construct a (non-generic) framework (G, p) with $c_{13} = 12$, $c_{16} = 23$, $c_{17} = 43$, $c_{18} = 47$, $c_{23} = 37$, $c_{25} = 13$, $c_{28} = 29$, $c_{34} = 17$, $c_{45} = 1$, $c_{46} = 11$,

$c_{57} = 31$, $c_{58} = 19$, $c_{67} = 5$. Substituting these values into (1), we may use a computer algebra package to eliminate $x_{12}, x_{14}, x_{35}, x_{56}$ to obtain a single polynomial equation for x_{15} of degree 68. This tells us that if we repeat the same procedure for a generic choice of (G_3, p), the polynomial equation we obtain for x_{15} will have degree at least 68 and will have distinct roots. Hence $\text{comp}(G_3) \geq 68$.

References

[1] D. N. Bernstein, The number of roots of a system of equations, *Funkcional. Anal. i Priložen* **9** (1975), 1–4.

[2] C. Borcea and I. Streinu, The number of embeddings of minimally rigid graphs, *Discrete Comput Geom* **31** (2004), 287–303.

[3] J. Capco, M. Gallet, G. Grassegger, C. Koutschan, N. Lubbes and J. Schicho, The number of realizations of a Laman graph, preprint arXiv 1701.05500v1.

[4] I.Z. Emiris and G. Moroz, The assembly modes of 11-bar linkages, in *Proc. IFToMM World Cong. Mechanism and Machine Sc.* Guanajuato, Mexico (2011).

[5] I.Z. Emiris, E.P. Tsigaridas and A. Varvitsiotis, Mixed volume and distance geometry techniques for counting Euclidean embeddings of rigid graphs, in *Distance geometry: Theory, Methods, and Appications* Springer, New York 2013 23–45.

[6] I.Z. Emiris and I. D. Psarros, Counting Euclidean embeddings of rigid graphs, preprint arXiv:1402.1484v2.

[7] Z. Fekete, T. Jordán and W. Whiteley, An Inductive Construction for Plane Laman Graphs via Vertex Splitting, in Algorithms - ESA 2004, *Lecture Notes in Comput. Sci.*, **3221** (2004), 299-310.

[8] S. Gortler, A. Healy, and D. Thurston, Characterizing generic global rigidity, *American J. Math.* **132** (2010) 897–939.

[9] S. Gortler and D. Thurston, Generic global rigidity in complex and pseudo-Euclidean spaces, in *Rigidity and Symmetry*, Fields Institute Communications **70**, Springer New York 2014 131–154.

[10] B. Hendrickson, Conditions for unique graph realisations, *SIAM J. Comput.* **21** (1992), 65–84.

[11] B. Jackson and T. Jordán, Connected rigidity matroids and unique realisations of graphs, *J. Combin. Theory Ser. B* **94** (2005), 1–29.

[12] B. Jackson, T. Jordán, and Z. Szabadka, Globally linked pairs of vertices in equivalent realizations of graphs, *Discrete and Computational Geometry* **35** (2006), 493–512.

[13] B. Jackson and J. C. Owen, The number of equivalent realisations of a rigid graph, preprint arXiv 1204.1228v2.

[14] G. Laman, On graphs and rigidity of plane skeletal structures, *J. Engrg. Math* **4** (1970), 331–340.

[15] J.B. Saxe, Embeddability of weighted graphs in k-space is strongly NP-hard, Tech. Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA, 1979.

[16] R. Steffens and T. Theobald, Mixed volume techniques for embeddings of Laman graphs, *Comp. Geom.: Theo. Appl.* **43** (2010)84–93.

Branching packing theorems in finite and infinite digraphs

Attila Joó

MTA-ELTE Egerváry Research Group
Department of Operations Research
Eötvös Loránd University
1117 Budapest, Pázmány P. sétány 1/C.,
Hungary
joapaat@cs.elte.hu

Abstract: In this paper we present a common generalization of the maximal arborescence packing theorem of Cs. Király [10] (which itself is a common generalization of the reachability based branching packing theorem [9] and a matroid based branching packing result [3]) and two of our earlier works about packing branchings in infinite digraphs, namely [6] and [7].

Keywords: branching packing, infinite graphs, matroids

Infinite graph theory has a great tradition in Hungary. It was one of the favourite topics of Paul Erdős, the most unique personality and mind in the history of Hungarian mathematics. Looking for infinite generalization of theorems in finite graph theory was a usual starting point of his works in this field (we refer to the survey [11]). We present here branching packing results for infinite digraphs.

A digraph D is called an **arborescence** rooted at r if it is a directed tree where each vertex is reachable from r by a directed path. Vertex-disjoint arborescences form a **branching**. The **root set** of a branching is the set of the roots of the constitutive arborescences. A **spanning branching** of a digraph is spanning subgraph of it which is a branching.

Theorem 1 (Edmonds' branching theorem, [4]) *Let $D = (V, A)$ be a finite digraph and let $R_i \subseteq V$ be nonempty for $i = 1, \dots, k$. There exists a system $\{\mathcal{B}_i\}_{1 \leq i \leq k}$ of pairwise edge-disjoint spanning branchings in D where the root set of $\mathcal{B}_i$ is R_i if and only if for every nonempty $X \subseteq V$ has at least as many ingoing edges as many R_i are disjoint from it.*

The theorem above fails for infinite digraphs. Indeed, R. Aharoni and C. Thomassen proved in [1] that there is no $k \in \mathbb{N}$ such that the k-edge-connectedness of an infinite graph implies the existence of even two edge-disjoint spanning trees. The first positive result that we know is the following.

Theorem 2 (C. Thomassen) *In Theorem 1 it is enough to assume, instead of the finiteness of D, that D does not contain backward-infinite paths.*

With entirely different methods we proved the following similar result.

Theorem 3 (A. Joó, [6]) *In Theorem 1 it is enough to assume, instead of the finiteness of D, that any forward-infinite path of D meets all the sets R_i.*

One of the new difficulties in the infinite case that the "reduce to a smaller problem or problems and use induction" approach fails because the resulting subproblems are no longer smaller in any suitable sense.

We investigated later the possibility of packing infinitely many branchings with prescribed nonempty root sets $\{R_i\}_{i \in \mathbb{N}}$. The literal generalization of the condition of Edmonds (with cardinals) does not even imply that any vertex v is simultaneously reachable by edge-disjoint directed paths from the root sets R_i (which would be obviously necessary). We proved the following theorem.

Theorem 4 (A. Joó, [7]) *Let $D = (V, A)$ be a digraph and let $R_i \subseteq V$ for $i \in \mathbb{N}$. Assume that any backward-infinite path in D meets all the sets R_i. Then there exists a system $\{\mathcal{B}_i\}_{i\in\mathbb{N}}$ of pairwise edge-disjoint spanning branchings in D where the root set of $\mathcal{B}_i$ is R_i if and only if for each $v \in V$ there exists a system of pairwise edge-disjoint directed paths $\{P_i^v\}_{i\in\mathbb{N}}$ in D such that P_i^v goes from R_i to v.*

We present an example which shows that one cannot replace "backward-infinite" by "forward-infinite" in the theorem above. Let $V = \{t\} \cup \{(m, n) \in \mathbb{N} \times \mathbb{N} : m \leq n\}$ and let A be the set of the following edges (see Figure 1)

1. infinitely many parallel edges from $(m, n + 1)$ to (m, n),
2. edge from (m, n) to $(m + 1, n)$,
3. edge from $(2m + 2, n)$ to $(2m, n)$,
4. edge from (m, m) to t,
5. edge from t to $(2m + 1, n)$ (*not in the figure*!).

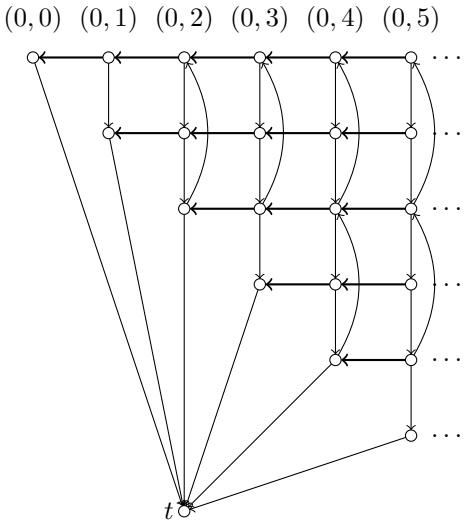


Figure 1: The outgoing edges of t (a single edge to each vertex in an odd row) are not on the figure because of transparency reasons. The thick horizontal edges stand for infinitely many parallel edges. Finally $R_n = \{(0, n)\}$.

Observe that after the deletion of t just finitely many vertices are reachable from any vertex which shows that there is no forward-infinite path in $D := (V, A)$. Let $R_n = \{(0, n)\}$. It is easy to check (using Figure 1) that path condition holds. Suppose to the contrary that there is a $\mathcal{B} = \{\mathcal{B}_n\}_{n\in\mathbb{N}}$ spanning branching packing. For $\mathcal{B}_0$ the only possibility to reach t is to use the single edge from $(0, 0)$ to t. Suppose that we already know for some $0 < N$ that $\mathcal{B}_n$ contains the path $P_n := (0, n), (1, n), \ldots, (n, n), t$ whenever $n < N$. By using just the remaining edges, t is no more reachable from columns $0, \ldots, N - 1$. Hence for $\mathcal{B}_N$ the path $(0, N), (1, N), \ldots, (N, N), t$ is the only possible option to reach t (see Figure 1). On the other hand after the deletion of the edges of paths P_n for all n the vertices $\{(0, n) : 1 \leq n \in \mathbb{N}\}$ are no longer reachable from $\{(0, 0), t\}$. This prevents $\mathcal{B}_0$ to be a spanning branching rooted at $(0, 0)$ which is a contradiction.

In the proof of Theorem 1 a vertex set X is usually called tight if it has exactly as many ingoing edges as many R_i are disjoint from it. The right generalization of this notion turned out to be the following.

For a vertex set X let $O(X) = \{i \in \mathbb{N} : R_i \cap X = \varnothing\}$. Then X is called tight if any system $\{P_i\}_{i \in O(X)}$ of edge-disjoint paths where P_i is a $R_i \to X$ path uses all the ingoing edges of X. In a proof of Theorem 1 we delete an appropriate outgoing edge e of R_1 and extend R_1 with $\mathsf{end}(e)$. Then we show that the resulting system still satisfies the condition an hence we are done by induction. Let us call a set X dangerous if it is tight and intersects R_1. In the finite case, it is routine to check that the outgoing edge e of R_1 is appropriate if and only if it does not enter any dangerous set X. This characterization of the suitable edges remains true in the infinite case but it is no longer trivial, in fact the proof is longer than the whole proof of Theorem 1. One cannot guarantee the existence of such an e without any restriction on the behaviour of the infinite directed paths. In the following example any e in question enters some dangerous set.

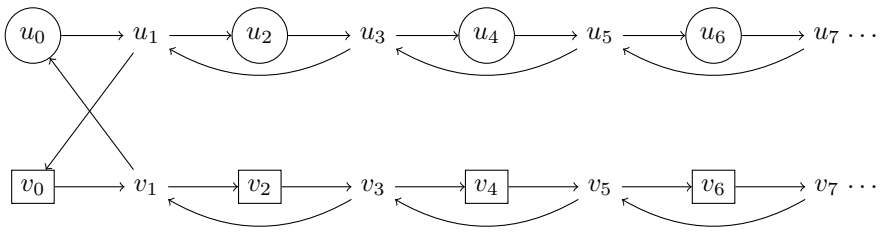


Figure 2: Elements of R_1 are circled and elements of R_2 are in rectangle.

The branching packing theorem of Edmonds has been generalized in several different directions in the finite case. Let us mention some of them.

Theorem 5 (N. Kamiyama, N. Katoh, A. Takizawa, [9]) *Let $D = (V, A)$ be a finite digraph and let $R_i \subseteq V$ for $i = 1, \ldots, k$. Then there exists a system $\{\mathcal{B}_i\}_{1 \leq i \leq k}$ of pairwise edge-disjoint branchings in D where the root set of $\mathcal{B}_i$ is R_i and $V(\mathcal{B}_i)$ consists of all the vertices which are reachable from R_i in D if and only if each nonempty $X \subseteq V$ has at least as many ingoing edges as many R_i are disjoint from X but can reach it by a directed path.*

Call a triple $\mathfrak{R} = (D, \mathcal{M}, \pi)$ a **matroid-rooted digraph** if $D = (V, A)$ is a digraph, $\mathcal{M} = (S, \mathcal{I})$ is a matroid and $\pi : S \to \mathcal{P}(V) \setminus \{\varnothing\}$. For $X \subseteq V$ let $\boldsymbol{\mathcal{S}(X)} = \{i \in S : \ \pi(i) \cap X \neq \varnothing\}$. The matroid-rooted digraph is called **independent** if $\mathcal{S}(v) \in \mathcal{I}$ for all $v \in V$. A **branching packing** $\mathcal{B}$ with respect to $\mathfrak{R}$ is a system of edge-disjoint branchings $\mathcal{B} = \{\mathcal{B}_i\}_{i \in S}$ in D where the root set of $\mathcal{B}_i$ is $\pi(i)$. A branching packing is called **total** if for each v the set $\{i \in S : v \in V(\mathcal{B}_i)\}$ is a base of $\mathcal{M}$.

Theorem 6 (O. Durand de Gevigney, V.-H. Nguyen, and Z. Szigeti, [3]) *Let $(D, \mathcal{M}, \pi)$ be a finite matroid-rooted digraph. Then there exists a total branching packing if and only if $(D, \mathcal{M}, \pi)$ is independent and each nonempty $X \subseteq V(D)$ has at least $r(\mathcal{M}) - r(\mathcal{S}(X))$ many ingoing edges in D.*

To formulate a common generalization of Theorems 5 and 6, let us denote by $\boldsymbol{\mathcal{N}(X)}$ (the **n**eed of X) the set spanned by the matroid elements i for which there is a $\pi(i) \to X$ path in D. The branching packing $\mathcal{B}$ is **maximal** if for all $v \in V$ the set $\{i \in S : v \in V(\mathcal{B}_i)\}$ is a base of $\mathcal{N}(v)$.

Theorem 7 (Cs. Király, [10]) *Let $(D, \mathcal{M}, \pi)$ be a finite matroid-rooted digraph. Then there exists a maximal branching packing if and only if $(D, \mathcal{M}, \pi)$ is independent and each $X \subseteq V(D)$ has at least $r(\mathcal{N}(X)) - r(\mathcal{S}(X))$ many ingoing edges in D.*

We investigated the possibility of the infinite generalization of the theorem above. To do so, we need to introduce the notion of infinite matroid. In the first half of the 20th century infinite matroids are most often defined like finite ones, with the additional axiom: an infinite set is independent if all of its finite subsets are independent. Let us mention vector spaces with linear independence and the graphic and transversal matroids for example. This approach was not a satisfactory infinite generalisation of

matroids since it spoils duality, one of the key features of finite matroid theory. Rado asked in 1966 for the development of a theory of infinite matroids with duality. This goal has been achieved in 2013 (see [2]). The "right" infinite generalization is the following. The pair $\mathcal{M} = (S, \mathcal{I})$ is a matroid if $\mathcal{I} \subseteq \mathcal{P}(S)$ and it satisfies the following axioms.

1. $\varnothing \in \mathcal{I}$,
2. $I \subseteq I' \in \mathcal{I}$ implies $I \in \mathcal{I}$,
3. if B is a $\subseteq$-maximal element of $\mathcal{I}$ and $I \in \mathcal{I}$ is not maximal, then there is an $i \in B \setminus I$ such that $(I \cup \{i\}) \in \mathcal{I}$,
4. if $I \in \mathcal{I}$ and $I \subseteq X \subseteq S$, then the set $\{I' \in \mathcal{I} : I \subseteq I' \subseteq X\}$ has a $\subseteq$-maximal element.

Now we turn to the generalization of Theorem 7. For an $I \in \mathcal{I}$ and $v \in V$, an **(I, v)-linkage** is a system of edge-disjoint directed paths $\{P_i\}_{i \in I}$ such that P_i goes from $\pi(i)$ to v. The maximality criteria leads to the following necessary condition beyond independence.

Condition 1 (linkage condition) *For all $v \in V$ there exists a base B_v of $\mathcal{N}(v)$ such that there is a (B_v, v)-linkage in D where B_v is a base of $\mathcal{N}(v)$.*

Indeed, a maximal branching packing contains such a linkage for any v. By applying the ideas of the proofs of theorems 3 and 4, we obtained the following.

Theorem 8 (A. Joó, [8]) *Let $(D, \mathcal{M}, \pi)$ be matroid-rooted digraph that satisfies one of the two conditions below. Then there exists a maximal branching packing if and only if $(D, \mathcal{M}, \pi)$ is independent and the linkage condition holds.*

- *$\mathcal{M}$ has finite rank and for any forward-infinite path P of D we have $\mathcal{N}(V(P)) \subseteq \mathsf{span}_{\mathcal{M}}(\mathcal{S}(V(P)))$*
- *$\mathcal{M}$ is a direct sum of countably many finite rank matroids and for any backward-infinite path P of D we have $\mathcal{N}(V(P)) \subseteq \mathsf{span}_{\mathcal{M}}(\mathcal{S}(V(P)))$.*

Q. Fortier, Cs. Király, M. Léonard, Z. Szigeti, and A. Talon introduced in [5], among other results, a further generalization of branching packing problems. Their idea is to use hyperdigraphs in which a directed edge has exactly one head and at least one tail. A hyperdigraph is called a branching if from the set of the vertices with zero indegree (the root of the branching) one can reach any vertex by a unique sequence of hyperedges. One can formulate all the theorems above in this generalized form and then reduce it to the original in the proof.

References

[1] R. Aharoni and C. Thomassen, *Infinite, highly connected digraphs with no two arc-disjoint spanning trees*, Journal of graph theory, 13 (1989), pp. 71–74.

[2] H. Bruhn, R. Diestel, M. Kriesell, R. Pendavingh, and P. Wollan, *Axioms for infinite matroids*, Advances in Mathematics, 239 (2013), pp. 18–46.

[3] O. Durand de Gevigney, V.-H. Nguyen, and Z. Szigeti, *Matroid-based packing of arborescences*, SIAM Journal on Discrete Mathematics, 27 (2013), pp. 567–574.

[4] J. Edmonds, *Edge-disjoint branchings*, Combinatorial algorithms, 9 (1973), pp. 91–96.

[5] Q. Fortier, Cs. Király, M. Léonard, Z. Szigeti, and A. Talon, *Old and new results on packing arborescences*, Tech. Report TR-2016-04, Egerváry Research Group, Budapest, 2016. `www.cs.elte.hu/egres`.

[6] A. Joó, *Edmonds' branching theorem in digraphs without forward-infinite paths*, Journal of Graph Theory, 83 (2016), pp. 303–311.

[7] A. Joó, *Packing countably many branchings with prescribed root-sets in infinite digraphs*, Journal of Graph Theory, (2016).

[8] A. Joó, *Independent and maximal branching packing in infinite matroid-rooted digraphs*, Tech. Report TR-2017-03, Egerváry Research Group, Budapest, 2017. `www.cs.elte.hu/egres`.

[9] N. Kamiyama, N. Katoh, and A. Takizawa, *Arc-disjoint in-trees in directed graphs*, Combinatorica, 29 (2009), pp. 197–214.

[10] Cs. Király, *On maximal independent arborescence packing*, SIAM Journal on Discrete Mathematics, 30 (2016), pp. 2107–2114.

[11] P. Komjáth, *Erdős's work on infinite graphs*, Erdős Centennial, 25 (2014), pp. 325–345.

Extremal problems and results in combinatorial rigidity

Tibor Jordán[1]

Department of Operations Research, Eötvös University, and
MTA-ELTE Egerváry Research Group on Combinatorial Optimization
Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
jordan@cs.elte.hu

Abstract: This note is a collection of questions and conjectures concerning various combinatorial properties of rigid and globally rigid graphs, along with some new results which provide motivation and partial solutions.

Keywords: rigid framework, rigid graph, globally rigid graph, rigidity matroid

1 Introduction

A d-dimensional (bar-and-joint) *framework* is a pair (G,p), where $G=(V,E)$ is a graph and p is a map from V to $\mathbb{R}^d$. We consider the framework to be a straight line *realization* of G in $\mathbb{R}^d$. Two realizations (G,p) and (G,q) of G are *equivalent* if $||p(u)-p(v)||=||q(u)-q(v)||$ holds for all pairs u,v with $uv\in E$, where $||.||$ denotes the Euclidean norm in $\mathbb{R}^d$. Frameworks (G,p), (G,q) are *congruent* if $||p(u)-p(v)||=||q(u)-q(v)||$ holds for all pairs u,v with $u,v\in V$.

We say that (G,p) is *globally rigid* in $\mathbb{R}^d$ if every d-dimensional realization of G which is equivalent to (G,p) is congruent to (G,p). The framework (G,p) is *rigid* if there exists an $\epsilon>0$ such that, if (G,q) is equivalent to (G,p) and $||p(v)-q(v)||<\epsilon$ for all $v\in V$, then (G,q) is congruent to (G,p). Intuitively, this means that if we think of a d-dimensional framework (G,p) as a collection of bars and joints where points correspond to joints and each edge to a rigid (i.e. fixed length) bar joining its end-points, then the framework is globally rigid if its bar lengths determine the realization up to congruence. It is rigid if every continuous motion of the joints that preserves all bar lengths must preserve all pairwise distances between the joints.

It is a hard problem to decide if a given framework is rigid or globally rigid. We obtain more tractable problems if we consider *generic frameworks* i.e. frameworks in which there are no algebraic dependencies between the coordinates of the vertices.

It is known that the rigidity of frameworks in $\mathbb{R}^d$ is a generic property, that is, the rigidity of (G,p) depends only on the graph G and not the particular realization p, if (G,p) is generic, see [22]. We say that the graph G is *rigid* in $\mathbb{R}^d$ if every (or equivalently, if some) generic realization of G in $\mathbb{R}^d$ is rigid. The problem of characterizing when a graph is rigid in $\mathbb{R}^d$ has been solved for $d=1,2$, and is a major open problem for $d\geq 3$.

A similar situation holds for global rigidity. Gortler, Healy and Thurston [6] proved that the global rigidity of d-dimensional frameworks is a generic property for all $d\geq 1$. We say that a graph G is *globally rigid* in $\mathbb{R}^d$ if every (or equivalently, if some) generic realization of G in $\mathbb{R}^d$ is globally rigid. Hendrickson

[1]This work was supported by the National Research, Development and Innovation Office, grant no. NKFIH K115483 and K 109240.

[7] proved two key necessary conditions for the global rigidity of a graph. We say that G is *redundantly rigid in* $\mathbb{R}^d$ if removing any edge of G results in a rigid graph.

Theorem 1 *[7] Let G be a globally rigid graph in $\mathbb{R}^d$. Then either G is a complete graph on at most $d+1$ vertices, or G is*
(i) $(d+1)$-connected, and
(ii) redundantly rigid in $\mathbb{R}^d$.

He conjectured that the necessary conditions of Theorem 1 together are also sufficient to imply the global rigidity of the graph in $\mathbb{R}^d$. It is indeed so for $d = 1, 2$. It is not hard to verifiy that a 1-dimensional generic framework (G, p) is globally rigid if and only if either G is the complete graph on at most two vertices or G is 2-connected. The characterization for $d = 2$ is as follows.

Theorem 2 *[10] Let G be a graph. Then G is globally rigid in $\mathbb{R}^2$ if and only if either G is a complete graph on at most three vertices, or G is 3-connected and redundantly rigid in $\mathbb{R}^2$.*

However, there exist counterexamples to his conjecture for $d \geq 3$, see [2, 15], and characterizing the globally rigid graphs in three-space and in higher dimensions remains another major open problem in rigidity theory. For the definitions not given here and for a detailed survey of rigid and globally rigid d-dimensional frameworks and graphs, and their applications, we refer the reader to [11, 14, 22].

2 A sufficient condition for global rigidity

In this section we prove that rigidity in $\mathbb{R}^{d+1}$ implies global rigidity in $\mathbb{R}^d$. We shall rely on the following results, due to Whiteley and Tanigawa, respectively.

Let $G = (V, E)$ be a graph. The *cone graph* of G, denoted by $G * u$, is obtained from G by adding a new vertex u and new edges (u, v) for all vertices $v \in V$.

Theorem 3 *[21] A graph G is rigid in $\mathbb{R}^d$ if and only if the cone graph of G is rigid in $\mathbb{R}^{d+1}$.*

We say that graph $G = (V, E)$ is *vertex-redundantly rigid* in $\mathbb{R}^d$ if $G - v$ is rigid in $\mathbb{R}^d$ for all $v \in V$.

Theorem 4 *[20] If G is vertex-redundantly rigid in $\mathbb{R}^d$ then it is globally rigid in $\mathbb{R}^d$.*

Theorem 5 *If G is rigid in $\mathbb{R}^{d+1}$ then it is vertex-redundantly rigid in $\mathbb{R}^d$.*

Proof: For a contradiction suppose that $G - v$ is not rigid in $\mathbb{R}^d$ for some vertex $v \in V$. It follows from Theorem 3 that the cone graph $(G - v) * u$ is not rigid in $\mathbb{R}^{d+1}$. Since G is a spanning subgraph of $(G - v) * u$, we obtain that G is not rigid in $\mathbb{R}^{d+1}$, a contradiction. □

For several extensions of Theorem 5 see [17]. Theorem 5 and Theorem 4 imply the desired sufficient condition.

Theorem 6 *If G is rigid in $\mathbb{R}^{d+1}$ then it is globally rigid in $\mathbb{R}^d$.*

The sufficient condition in Theorem 6 is not necessary: consider for example the cycles of length at least four in $\mathbb{R}^1$ and their iterated cone graphs.

3 Making rigid graphs globally rigid

We can use Theorem 6 to solve an extremal question about the size of a smallest augmenting set of edges that makes a rigid graph globally rigid.

Theorem 7 *Let $G = (V, E)$ be a rigid graph in $\mathbb{R}^d$ with $|V| \geq d + 1$. Then G can be made globally rigid in $\mathbb{R}^d$ by adding at most $|V| - d - 1$ edges.*

PROOF: We may suppose that G is minimally rigid in $\mathbb{R}^d$. Thus E is independent in the d-dimensional rigidity matroid, which implies that E is independent in the $(d + 1)$-dimensional rigidity matroid, too. Extend G to a minimally rigid graph $G' = (V, E + F)$ in $\mathbb{R}^{d+1}$. The size of the set F of the added edges can be obtained as follows:

$$|F| = (d+1)|V| - \binom{d+2}{2} - (d|V| - \binom{d+1}{2}) = |V| - (\binom{d+2}{2} - \binom{d+1}{2}) = |V| - d - 1. \quad (1)$$

Since G' is globally rigid in $\mathbb{R}^d$ by Theorem 6 the theorem follows. □

The upper bound in Theorem 7 is tight for all $d \geq 1$ and all $|V| \geq d + 1$: consider the graph G obtained from the complete graph K_d by adding $|V| - d$ vertices of degree d so that each vertex is fully connected to the complete subgraph K_d. It is easy to see that at least $|V| - d - 1$ new edges must be added in order to make G globally rigid.

3.1 Augmenting braced triangulations

By the following recent result of Jordán and Tanigawa we can improve on the upper bound of Theorem 7 for braced triangulations. We say that a graph is a *braced triangulation* if it contains a planar triangulation (i.e. a maximal planar graph) as a spanning subgraph.

Theorem 8 *[16] A braced triangulation is globally rigid in $\mathbb{R}^3$ if and only if it is 4-connected.*

Thus the augmentation problem boils down to finding a smallest set of new edges whose addition makes a braced triangulation 4-connected. It turns out that a simple characterization of the optimum (and a polynomial time algorithm for finding an optimal solution) follows from a result of Jackson and Jordán [10].

Let $H = (V, E)$ be a braced triangulation. We say that $D \subset V(H)$ is a *fragment* of H if there is a three-separator S for which D is the vertex set of a connected component of $H - S$. Since H is 3-connected and *4-independence free*[1] the min-max result from [10, Theorem 3.12] applies and gives that the size of a smallest set of new edges which makes H 4-connected is equal to

$$\max\{b(H) - 1, \lceil t(H)/2 \rceil\}, \quad (2)$$

where $b(H)$ denotes the maximum number of connected components of $H - S$ over all three-separators S of H and $t(H)$ is the maximum number of pairwise disjoint fragments in H. Since $H - S$ has exactly two connected components for every three-separator S of a braced triangulation (see [16]), we have $b(H) \leq 2$. Thus (2) reduces to $\lceil t(H)/2 \rceil$. It is not hard to show that in a (braced) triangulation the fragments form a laminar family and hence the minimal fragments are pairwise disjoint. Therefore $t(H)$ equals the number of minimal fragments of H.

By using this fact one can show that a triangulation on n vertices has at most (roughly) $\frac{2n}{3}$ minimal fragments[2]. Thus there is always an augmenting set of size at most $\frac{n}{3}$.

[1] A 3-connected graph H is called 4-*independence free* if there is no three-separator S in H which contains a fragment. Since the three-separators in a (braced) triangulation induce complete subgraphs, see e.g [16], every braced triangulation is 4-independence free.

[2] The worst case occurs in the family of triangulations which are obtained from another tringulation by inserting a vertex of degree three into each face.

4 Minimally globally rigid graphs

A graph G is *minimally globally rigid* in $\mathbb{R}^d$ if it is globally rigid but removing any edge from G leaves a graph which is not globally rigid in $\mathbb{R}^d$. We conjecture that the number of edges of a minimally globally rigid graph is linear in the number of vertices for every fixed d, and hence the minimum degree can be bounded by a (linear) function of d. It is well known that the minimally rigid graphs in $\mathbb{R}^d$ satisfy both of these properties.

Conjecture 9 *Let $G = (V, E)$ be minimally globally rigid in $\mathbb{R}^d$ with $|V| \geq d + 1$. Then*
(a) $|E| \leq (d + 1)|V| - \binom{d+2}{2}$ and
(b) the minimum degree of G is at most $2d + 1$.

Note that (a) implies (b). The upper bounds on the edge number and the minimum degree would be (close to being) tight for all d. Consider the complete bipartite graph $K_{d+1,|V|-d-1}$ with $|V| \geq \binom{d+1}{2}+d+2$. This graph can be obtained from K_{d+2} by a sequence of 1-extensions, and hence it is globally rigid in $\mathbb{R}^d$. Since every edge is incident with a vertex of degree $d + 1$, it is a minimally globally rigid graph with $(d+1)|V| - (d+1)^2$ edges. It may be interesting to construct minimally globally rigid graphs with minimum degree $2d + 1$.

In the rest of this subsection we show that Conjecture 9 is true for $d = 1, 2$. The one-dimensional case is easy to deduce from the fact that a graph is globally rigid in $\mathbb{R}^1$ if and only if it is 2-connected. By a result of Mader [19] a minimally 2-connected graph $G = (V, E)$ satisfies $|E| \leq 2|V| - 4$, unless G is a triangle.

We next verify the conjecture in the two-dimensional case. We need the following inductive construction of globally rigid graphs in $\mathbb{R}^2$. The *1-extension* operation removes some edge uw from the graph and adds a new vertex v as well as new edges vu, vw, vz for some vertex z different from u, w.

Theorem 10 *[10] Let $G = (V, E)$ be a graph with $|V| \geq 4$. Then G is globally rigid in $\mathbb{R}^2$ if and only if G can be obtained from K_4 by a sequence of edge additions and 1-extensions.*

Theorem 11 *Suppose that $G = (V, E)$ is minimally globally rigid in $\mathbb{R}^2$ with $|V| \geq 4$. Then $|E| \leq 3|V|-6$ and the minimum degree of G is equal to 3.*

Proof: Consider a sequence of graphs $G_1, G_2, ..., G_t$ for which $G_1 = K_4$, $G_t = G$, and G_i is obtained from G_{i-1} by an edge addition or 1-extension for all $2 \leq i \leq t$. Such a sequence exists by Theorem 10. Note that K_4 satisfies $|E| = 3|V| - 6$. Since G is minimally globally rigid, every edge addition operation used in this sequence adds an edge which will be split into two edges later by a 1-extension operation. This leads to a pairing, that is, a bijection between the added edges and a subset of the 1-extension operations. Each pair increases the number of vertices by one and the number of edges by three. A 1-extension operation alone increases the number of vertices by one and the number of edges by two. Thus, since K_4 satisfies $|E| = 3|V| - 6$, and the total number of edges added by the operations is not more than three times the number of added vertices, $G_t = G$ satisfies $|E| \leq 3|V| - 6$, as required.

The second part of the claim follows from the fact that, by the minimality of G, the graph G_t is obtained from G_{t-1} by a 1-extension operation. □

Consider a minimally globally rigid graph G in $\mathbb{R}^2$ which is vertex-redundantly rigid. Then G is minimally vertex-redundantly rigid by Theorem 4 and hence we can use the following result of Király and Kaszanitzky to prove that G satisfies the bounds of Conjecture 9.

Theorem 12 *[17] Let $G = (V, E)$ be a minimally vertex-redundantly rigid graph in $\mathbb{R}^d$. Then $|E| \leq (d + 1)|V| - \binom{d+2}{2}$.*

Theorem 12 is an immediate corollary of the following stronger result: if G is minimally vertex-redundantly rigid in $\mathbb{R}^d$ then every edge of G is a bridge in $\mathcal{R}_{d+1}(G)$, see [17, Lemma 6].

There exist globally rigid graphs which are not vertex-redundantly rigid in $\mathbb{R}^d$ for all $d \geq 2$. Still we beleive that a similar approach can be used to handle these graphs. The truth of the next conjecture would imply Conjecture 9.

Conjecture 13 *Let G be globally rigid in $\mathbb{R}^d$ and suppose that $G-e$ is not. Then e is a bridge in $\mathcal{R}_{d+1}(G)$.*

We have the following partial results (c.f. Theorem 1).

Theorem 14 *Let $G=(V,E)$ be $(d+1)$-connected and redundantly rigid in $\mathbb{R}^d$. Suppose that for some edge $e=uv\in E$ we have that either*
(i) $G-e$ is not $(d+1)$-connected, or
(ii) $G-e$ is not redundantly rigid in $\mathbb{R}^d$.
Then e is a bridge in $\mathcal{R}_{d+1}(G)$.

PROOF: (i) Suppose that e belongs to an M-circuit H of $\mathcal{R}_{d+1}(G)$. By [9, Lemma 2.5] this implies that $G-e$ contains $(d+1)$ pairwise openly-disjoint uv-paths. It is easy to see that in this case $G-e$ must be $(d+1)$-connected, a contradiction.

(ii) If $G-e$ is not redundantly rigid in $\mathbb{R}^d$ then there is an edge $f\in E$ for which $G-e-f$ is not rigid in $\mathbb{R}^d$. Then e is a bridge in $\mathcal{R}_d(G-f)$. Let w be an end-vertex of f which is disjoint from e. If e is not a bridge in $\mathcal{R}_d(G-v)$ then it belongs to some M-circuit H. But H is an M-circuit in $G-f$, too, contradicting the fact that e is a bridge in $\mathcal{R}_d(G-f)$. Hence e is a bridge in $\mathcal{R}_d(G-v)$. Therefore, by Theorem 3, e is a bridge in $\mathcal{R}_{d+1}((G-v)*v)$ and also in $\mathcal{R}_{d+1}(G)$. □

In this context one may go up one dimension and focus on rigidity rather than global rigidity. This motivates the search for upper bounds on the edge number of graphs satisfying special rigidity properties.

Lemma 15 *Let G be minimally globally rigid in $\mathbb{R}^d$. Then*
(i) every rigid subgraph of G in $\mathbb{R}^{d+1}$ is minimally rigid in $\mathbb{R}^{d+1}$, and
(ii) every M-circuit of G in $\mathbb{R}^{d+1}$ is non-rigid in $\mathbb{R}^{d+1}$.

PROOF: The proof follows from Theorem 6. It is based on the fact that replacing a rigid (resp. globally rigid) subgraph of a graph by another rigid (resp globally rigid) subgraph, on the same vertex set, preserves rigidity (resp. global rigidity). □

This leads to the following problem, which is interesting only if $d\geq 3$ (for otherwise the solution is straightforward, since every M-circuit is rigid in $\mathbb{R}^1$ and $\mathbb{R}^2$).

Problem 16 *Let $G=(V,E)$ be a graph and let $d\geq 3$ be a fixed dimension. Give a (tight) upper bound on $|E|$, in terms of d and $|V|$, if G satisfies that*
(i) every rigid subgraph of G is minimally rigid,
(ii) every M-circuit of G is non-rigid.

Clearly, (i) implies (ii). To see that the other implication does not always hold consider a rigid graph G in $\mathbb{R}^3$ obtained from the well-known double banana graph by adding an edge. This graph satisfies (ii), but does not satisfy (i). More generally, consider a chain of K_5's, obtained by taking 2-sums, and add some more edges (bridges) to make it rigid in $\mathbb{R}^3$. This graph satisfies (ii) and has roughly $\frac{10}{3}|V|-6$ edges.

5 Squares of graphs and connectivity

The *square* G^2 of a graph G is obtained from G by adding a new edge uv for all non-adjacent vertex pairs u,v of G with a common neighbour. Squares of graphs (sometimes called molecular graphs) can be used as a model in the study of rigidity properties of molecules in three-space. The characterization of globally rigid squares in $\mathbb{R}^3$ is not known. We recall the following conjecture due to Connelly, Jordán, and Whiteley.

Let $H=(V,E)$ be a multigraph. We say that H is *highly m-tree-connected* if $H-e$ contains m edge-disjoint spanning trees for all $e\in E$.

Conjecture 17 *[4] Suppose that G has no cycles of length at most four. Then G^2 is globally rigid in $\mathbb{R}^3$ if and only if G^2 is 4-connected and the multi-graph $5G$ is highly 6-tree-connected.*

Note that the original conjecture in [4] was slightly different and incomplete: it did not include the 4-connectivity condition and the "only if" implication. The following example from [12] shows that it is necessary to exclude short cycles: consider two four-cycles with a common vertex. For this graph G we have that $5G$ is highly 6-tree-connected but G^2 is not even redundantly rigid in $\mathbb{R}^3$. If we replace the four-cycles by five-cycles, we obtain a graph for which $5G$ is highly 6-tree-connected and G^2 is redundantly rigid. However, G^2 is not 4-connected and hence it is not globally rigid in $\mathbb{R}^3$.

Lovász and Yemini [18], resp. Connelly, Jordán, and Whiteley [4] conjectured that every sufficiently highly connected (in terms of d) graph is rigid (resp. globally rigid) in $\mathbb{R}^d$. These conjectures (which are closely related by Theorem 4) are open for all $d \geq 3$.

The former conjecture was verified for squares of graphs.

Theorem 18 *[13] Suppose that G^2 is 7-connected for some graph G. Then G^2 is rigid in $\mathbb{R}^3$.*

The next target might be a similar result for globally rigid squares in three-space. Currently we can only verify the following rather special case.

Theorem 19 *Let G be a graph and let $k \geq 8$ be an integer. Suppose that G^2 is k-connected and the maximum degree of G is at most $k-7$. Then G^2 is globally rigid in $\mathbb{R}^3$.*

Proof: By Theorems 4 and 18 it suffices to show that $H = (G-v)^2$ is 7-connected for all $v \in V$. For a contradiction suppose that there is vertex separator S of size six in H. Note that for every component D of $H - S$ there is at least one edge from v to D in G, for otherwise $S + v$ is a separator in G^2 of size at most 7. Moreover, since H has at least k vertices and v has degree at most $k-7$, there is a component D' of $H - S$ which contains a vertex w with $vw \notin E(G)$. Now $S \cup (N_G(v) \cap V(D')) \cup \{v\}$ is a separator in G of size at most $6 + k - 7 - 1 + 1 \leq k - 1$, a contradiction. □

In particular, if G has maximum degree 4 and G^2 is 11-connected then G^2 is globally rigid.

The investigation of globally rigid powers of graphs was initiated by Cheung and Whiteley [1]. Among other results they proved that G^{d+1} is globally rigid in $\mathbb{R}^d$ if and only if G is connected. For G^d we have the following conjecture. We say that a path P from vertex u_1 to vertex u_2 is *separating* in a connected graph G if G can be obtained from two disjoint non-trivial connected graphs G_1, G_2, with $u_i \in V(G_i)$, $i = 1, 2$, by adding the internal vertices and edges of P. For example, a separating path on two vertices corresponds to a bridge e of G for which each component of $G - e$ has at least two vertices.

Conjecture 20 *Let G be a connected graph and $d \geq 2$. Then G^d is globally rigid in $\mathbb{R}^d$ if and only if G does not contain a separating path on d vertices.*

A proof of Conjecture 20 up to $d = 3$ was announced in [1].

6 Minimum cost globally rigid spanning subgraphs

The *Minimum cost globally rigid spanning subgraph problem* is as follows: given a graph $G = (V, E)$, a cost function $c : E \to \mathbb{R}$, and a positive integer d, find a spanning subgraph $H = (V, E')$ which is globally rigid in $\mathbb{R}^d$ and for which $c(E') = \sum_{e \in E'} c(e)$ is as small as possible. If c is uniform, we look for a *minimum size* globally rigid spanning subgraph. In the metric version of the problem G is a complete graph and c satisfies the triangle inequality. Another interesting special case is when $c(e) \in \{0, 1\}$ for all $e \in E$. This version is called the *minimum size globally rigid augmentation problem*. We may obtain similar problems by changing global rigidity to redundant rigidity (or any other rigidity property) above.

Most of the existing results concerning this problem deal with the 1-dimensional case and are formulated in terms of 2-connected or 2-edge-connected spanning subgraphs. Garcia and Tejel [5] solve the

redundantly rigid augmentation problem for $d = 2$ for minimally rigid input graphs and show that this version is NP-hard for general graphs.

Since global rigidity is equivalent to 2-connectivity in $\mathbb{R}^1$, the (uniform cost version of the) 1-dimensional case of this problem contains the Hamilton cycle problem as a special case. By iterated coning we can add $d-1$ additional vertices and 0-cost edges and reduce the 1-dimensional case to the d-dimensional case for any given d. This shows that the problem is NP-hard for all $d \geq 1$, even for 0-1-valued cost functions. A similar argument works for redundant rigidity, too.

This leads to a large family of optimization problems for which one should look for approximation algorithms.

Problem 21 *Design efficient approximation algorithms for the Minimum cost globally rigid spanning subgraph problem.*

The low dimensional cases are also quite interesting. Note that testing feasibility is a difficult open problem for $d \geq 3$.

Since a redundantly rigid graph in $\mathbb{R}^2$ on vertex set V has at least $2|V| - 2$ edges, Theorem 1 and Theorem 11 imply the following:

Theorem 22 *There is a 1.5-approximation algorithm for the minimum size globally rigid spanning subgraph problem in $\mathbb{R}^2$.*

A similar result [14, Theorem 3.6.4] can be used to obtain a 1.5-approximation algorithm for the redundantly rigid version.

7 Acknowledgements

We thank Csaba Király for several useful comments.

References

[1] M. Cheung and W. Whiteley: Transfer of global rigidity results among dimensions: graph powers and coning, preprint, 2008.

[2] R. Connelly: On generic global rigidity, in *Applied Geometry and Discrete Mathematics*, DIMACS Ser. Discrete Math, Theoret. Comput. Sci. 4, AMS, 1991, pp 147-155.

[3] R. Connelly: Generic global rigidity, *Discrete Comp. Geometry* 33 (2005), pp 549-563.

[4] R. Connelly, T. Jordán, and W. Whiteley: Generic global rigidity of body-bar frameworks, *J. Combinatorial Theory, Ser. B.*, Vol. 103, Issue 6, November 2013, pp. 689-705.

[5] A. Garcia and J. Tejel: Augmenting the Rigidity of a Graph in $\mathbb{R}^2$, *Algorithmica*, February 2011, Volume 59, Issue 2, pp 145-168.

[6] S. Gortler, A. Healy, and D. Thurston: Characterizing generic global rigidity, *American Journal of Mathematics*, Volume 132, Number 4, August 2010, pp. 897-939.

[7] B. Hendrickson: Conditions for unique graph realizations, *SIAM J. Comput* 21 (1992), pp 65-84.

[8] B. Jackson and T. Jordán: Independence free graphs and vertex-connectivity augmentation, *J. Combinatorial Theory*, Ser. B., Vol. 94, 31-77, 2005.

[9] B. Jackson and T. Jordán: The Dress conjectures on rank in the 3-dimensional rigidity matroid, *Advances in Applied Mathematics*, Vol. 35, 355-367, 2005.

[10] B. Jackson and T. Jordán: Connected rigidity matroids and unique realization graphs, *J. Combin. Theory Ser. B* 94, 2005, pp 1-29.

[11] B. Jackson and T. Jordán: Graph theoretic techniques in the analysis of uniquely localizable sensor networks, in: G. Mao, B. Fidan (eds), Localization algorithms and strategies for wireless sensor networks, IGI Global, 2009, pp. 146-173.

[12] B. Jackson and T. Jordán: Rigid components in molecular graphs, *Algorithmica*, Vol. 48, No. 4 (2007) 399-412.

[13] T. Jordán: Highly connected molecular graphs are rigid in three dimensions, *Information Proc. Letters*, Vol. 112, 2012, pp. 356-359.

[14] T. Jordán: Combinatorial rigidity: graphs and matroids in the theory of rigid frameworks, in: Discrete Geometric Analysis, MSJ Memoirs, vol. 34, pp. 33-112, 2016.

[15] T. Jordán, C. Király, and S. Tanigawa: Generic global rigidity of body-hinge frameworks, *J. Combinatorial Theory, Ser. B.*, Vol. 117, 59-76, 2016.

[16] T. Jordán and S. Tanigawa: Global rigidity of triangulations with braces, preprint, April 2017.

[17] C. Király and V. Kaszanitzky: On minimally highly vertex-redundantly rigid graphs, *Graphs and Combinatorics* (2016) 32:225-240.

[18] L. Lovász and Y. Yemini: On generic rigidity in the plane, *SIAM. J. on Algebraic and Discrete Methods*, 3(1), 91???98, 1982.

[19] W. Mader: Über minimal n-fach zusammenhängende, unendliche Graphen und ein extremal Problem, *Arch. Math.* 23, 553-560 (1972).

[20] S. Tanigawa: Sufficient conditions for the global rigidity of graphs, *J. Combin. Theory Ser. B* 113, 2015, pp 123-140.

[21] W. Whiteley: Cones, infinity and one-story buildings, *Structural Topology* 8 (1983), pp. 53-70.

[22] W. Whiteley: Some matroids from discrete applied geometry, in Matroid theory (J.E. Bonin, J.G. Oxley and B. Servatius eds., Seattle, WA, 1995), Contemp. Math., 197, Amer. Math. Soc., Providence, RI, 1996, 171–311.

A Primal-Dual Approach for Large Scale Integer Problems

Alpár Jüttner

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
alpar@cs.elte.hu

Péter Madarasi

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
madarasi@cs.elte.hu

Abstract: This paper presents a refined approach to using column generation to solve specific type of large integer problems. A primal-dual approach is presented to solve the Restricted Master problem belonging to the original optimization task. Firstly, this approach allows a faster convergence to the optimum of the LP relaxation of the problem. Secondly, the existence of both an upper and lower bound of the LP optimum at each iteration allows a faster searching of the Branch-and-Bound tree. To achieve this an *early termination* approach is presented. The technique is demonstrated on the Generalized Assignment problem and Parallel Machine Scheduling problem as two reference applications.

Keywords: large scale optimization, column generation, primal-dual methods, integer programming, scheduling

1 Introduction

One of the most successful approaches to solve large scale practical combinatorial optimization problems is the combination of special linear programming techniques such as Dantzig-Wolfe Decomposition, Column Generation or Lagrangian relaxation with Cutting Planes, Branch-and-Bound (B&B) or certain iterative rounding techniques. Methods of this type are collectively known as *Branch-and-Cut-and-Price (B-C-P)*.

These approaches assume that the problem to be solved is formulated as a huge but well structured linear (or integer) program (often referred to as a *master problem (MP)*), which is then decomposed into a higher and a lower level subproblem, referred to as *restricted master problem (RMP)* and *column generator (CG)* or *pricing problem*. In case of Lagrangian relaxation, they are called *Lagrangian dual problem* and *Lagrangian subproblem*, respectively [13].

Beginning with the early results of Ford and Fulkerson [12], Appelgren[2], and others, especially after high performance linear programming solvers became widely available, Column Generation and Branch-and-Price are now standard tools for tackling various industrial math optimization problems. [3, 22, 20, 19]. It has successfully been applied to versions of traveling salesman, vehicle routing and crew scheduling problems [6], airline crew pairing [5], scheduling and fleet assignments, in telecommunication (network dimensioning, resource management and routing) and to staff scheduling problems [8], as well as to generic combinatorial optimization problems such as (integer) multicommodity flows [12], maximum stable-set [4] and graph coloring problems [16]. These works made extensive efforts on improving convergence of Column Generation and on developing efficient problem specific branching strategies (see Sections 2 and 3). On the other hand, several problem classes are still practically intractable, even though they seem to fit well into this framework.

The aim of this work is to discover ways to further widen the applicability of this approach by presenting a novel primal-dual solution technique that in one hand provides a faster convergence of the LP relaxation of the problem and in the other hand, allows a more effective execution of the usual Branch-and-Bound scheme to find the integer optimum solution.

The rest of the paper is organized as follows. Section 2 presents a refined approach that improves on the convergence rate solving the RMS problem in practice and, in addition, is able to provide both a lower and an upper bound of optimum at each iteration. Then, utilizing this property, Section 3 presents the *early termination* technique for B&B in order to speed up finding the integer optimum. Finally Section 4 presents the applicability of the proposed approach to two specific well-know optimization problems.

2 Primal-dual method for solving the RMS

When implementing a column generation based solution, one must often face the poor convergence of the simplex-based RMP, especially towards the end of the computation. A close to optimal solution may be found relatively fast, but then a long time is needed to find the real optimum (called the *tail-off effect*). Another related phenomenon is the heavy oscillation of the dual variables instead of a smooth convergence to the optimal values. This is widely considered as the main reason for the poor performance [14]. One of the first proposals for handling this issue is the BOXSTEP method proposed by Marsten at al. [15] and the *Stabilized Column Generation* proposed by du Merle et al. [9], which is considered the most promising stabilization technique.

Although these techniques are based on the dual considerations of the RMP, they are still *primal approaches* from the perspective of the Master Problem, with the property that they maintain a (non-optimal) feasible solution during the execution.

On the other hand, Lagrangian relaxation [10, 13, 11, 23, 19] represents a completely different approach. The Lagrangian subproblem computes a lower bound to the optimum of the Master Problem, while the Lagrangian dual problem aims at finding the parameters maximizing the lower bound, which maximum is in fact equal to the optimum of the (linear relaxation of) the Master Problem. The Lagrange relaxation[13] of the same problem combined with the standard subgradient method often provides a rapidly converging *lower bound*, while requires solving the same pricing subproblem. Unfortunately, Lagrange relaxation alone cannot produce a primal feasible solution, and the subgradient method, while it is very simple to implement still in many cases converges extremely fast, also suffers from frequent instability, tends to "stuck" and fails to eventually find the optimum.

Therefore, we propose a combination of the subgradient method with a primal approach. For the latter one we chose a linear-programming based stabilization [17]. This technique do not use the dual solution of the master problem as the price vector for column generation, but combines it with the preceding dual solutions. The smoothing rule proposed in [21], and reconsidered in [17] suggests $\tilde{\pi}^t = \alpha\hat{\pi} + (1-\alpha)\pi^t$ using as pricing vector, where π^t denotes the current dual vector, $\hat{\pi}$ is the incumbent dual vector and $\alpha \in [0, 1)$. In [17] proposes an efficient self-adjusting scheme adapting α to the phases of the algorithm.

Although the subgradient method convergences highly effectively, its instability and occasional divergence makes it impractical in case of large problems.

We improve this method be periodically inserting subgradient-based improving phases into the above primal algorithms. The initial step size of subgradient phase is calculated from the average oscillation of $(||\hat{\pi} - \pi^t||)$ of the last primal steps and it stops when no improvents is found within a constant number of streps. In this way a steady convergence of both the lower and upper bound can be ensu

3 Branching Strategies

It is well known that the branching scheme used in the conventional Branch-and-Bound method does not applicable for column generation since it would require excluding certain solutions from the pricing subproblem. Instead, various alternative branching schemes have been proposed. They are rather problem specific and partition the integer solutions of the problem in a way that is compatible with the pricing subproblem. See e.g. [7, 1, 18] for some illustrative examples. Section 4 presents such branching rules for two specific problems.

Even though the primal-dual approach above realize a considerable speedup, we still cannot afford running the column generation up to finding LP optimum at each node of the branch tree. Exploiting

the fact that the primal-dual approach maintains both a lower and an upper bound converging to the optimal value, two ideas are proposed for *early termination* of the solutions of the RMS subproblem.

Early cut. Normally, a node of the B&B tree is pruned when either the LP subproblem belonging to the node is infeasible or its LP optimum is worse than the best integer solution found so far. However, the existence of a lower bound to the LP optimum at each iterations allows us to terminate the solution as soon as the lower bound reaches the cost of the best integer.

Early branching. When solving the RMP, we iteratively generate an increasing subset of columns of the full problem and calculate the best LP solution obtainable using only those columns. In vast majority of the cases these solutions are fractional. Therefore as soon as the LP solution of the RMP goes below the best integer solution so far, we can conclude that branching will be inevitable. Thus we stop generating further columns and branch immediately.

These techniques significantly reduce the time required to process one node of the B&B tree, while — if properly implemented — it increases the size of the B&B tree only marginally.

4 Reference Applications

4.1 Generalized Assignment

In the generalized assignment problem we are given n jobs to be assigned to m agents. Each agent i has capacity u_i, and when job j is assigned to agent i, it requires capacity d_{ij} and costs c_{ij}. The solution consists of matching each job to exactly one agent, so that the capacities of the agents are respected and the total assignment cost is minimized.

Let $K_i = \{x_1^i, x_2^i, \ldots, x_{k_i}^i\}$ be the set of all feasible assignment of jobs to agent i, that is $x_k^i = (x_{k1}^i, x_{k2}^i, \ldots, x_{kn}^i)$ satisfies

$$\sum_{1 \le j \le n} d_{ij} x_{kj}^i \le u_i \tag{1}$$

$$x_{kj}^i \in \{0,1\} \ (j = 1, \ldots, n). \tag{2}$$

Let $z_k^i \in \{0,1\}$ $(i = 1, \ldots, m, \quad k \in K_i)$ indicate whether assignment x_k^i is selected for agent i. Using these notations, the generalized assignment problem can be formulated as follows.

$$\min \sum_{1 \le i \le m} \sum_{1 \le k \le k_i} z_k^i \sum_{1 \le j \le n} c_{ij} x_{kj}^i \tag{3}$$

$$\sum_{1 \le i \le m} \sum_{1 \le k \le k_i} z_k^i x_{kj}^i = 1 \ (j = 1, \ldots, n) \tag{4}$$

$$\sum_{1 \le k \le k_i} z_k^i \le 1 \ (i = 1, \ldots, m) \tag{5}$$

$$z_k^i \in \{0,1\} \ (i = 1, \ldots, m, \quad k \in K_i), \tag{6}$$

where the first set of constraints provides that each job is assigned to exactly one agent, while the second one enforces that at most one feasible assignment is chosen for all the agents.

The corresponding pricing problem consists of finding a feasible assignment to one of the agents with minimum reduced cost, which can be reformulated as a binary knapsack problem.

For this problem, the following two branching rules are used. In each node, we fix an agent i and job j, and create two subproblems (a) job j must be assigned to agent i (b) job j is not allowed to be assigned to agent i. Thus in each node we are given a subproblem, where some agent-job pairs are bounded and other ones are forbidden. All these restrictions can easily be incorporated to the knapsack problem, and the columns representing forbidden assignments can be avoided.

4.2 Parallel Machine Scheduling

In this problem, jobs $J := \{1, \ldots, n\}$ are given with processing times p_j, due times d_j and weights w_j. These jobs are to be processed by m identical machines while minimizing $\sum_{j=1}^{n} w_j \max(0, C_j - d_j)$, where C_j is the completion time of job j.

A *schedule* of a single machine is an $s = (j_1, j_2, \ldots, j_k)$ sequence of jobs which induces completion times $C_{j_i} = \sum_{l=0}^{i} p_{j_l}$ and costs $c(s) = \sum_{i=1}^{k} w_{j_i} \max(0, C_{j_i} - d_{j_i})$ The column generation formulation of this problem consists of the set of variables x_k^s for each machine $k = 1, \ldots, m$ and for each possible s_k schedule of this machine. The formulation is as follows.

$$\min \sum_{k=1}^{m} \sum_{s \in S_k} c(s) x_s \tag{7}$$

$$\sum_{k=1}^{m} \sum_{s \in S_k} \chi_s(i) x_s = 1 \quad \forall i = 1, \ldots, j \tag{8}$$

$$\sum_{s \in S_k} x_s = 1 \quad \forall k = 1, \ldots, m \tag{9}$$

$$x_s \in \{0, 1\} \quad \forall k = 1, \ldots, m, \quad s \in S_k \tag{10}$$

Where χ_s is the characteristic vector of s, i.e. $\chi_s(i) := |\{j \in s : j = i\}|$.

The corresponding pricing subproblem consists of finding a schedule for a single machine with an additional constant "price" y_j of processing job j. In order to make it solvable by a standard dynamic programming approach[17], we also allow multiple processing of a job by a single machine, but limit the maximum number of jobs processed by a single machine to be at most n.

To obtain an integer solution we apply a branch and bound method with the following branching rule.

In every node of the branching tree, for each machine k, we specify a series of subsets J_k^i of allowed jobs as the i^{th} job to be processed. This problem can also be solved using standard dynamic programming technique in time $O(n^2 T)$, by calculating the values

$$c(t, l) := \begin{cases} +\infty & \text{if } t < 0, \\ 0 & \text{if } k = 0, t = 0, \\ \min\limits_{j \in J_k^l} \left(w_j \max(0, t - d_j) - y_j + c(t - p_j, l - 1) \right) & \text{otherwise} \end{cases} \tag{11}$$

for each values of $k = 0, \ldots, n$ and $t = 0, \ldots, T$. The computed value $c(t, l)$ is the cost of the optimal sequence consisting of j jobs with the last job finished in time t.

In the root node of the branching tree we set $J_k^i := J$ for all k and i. Then we apply two different kind of branchings.

1. If a job j appears in the (fractional) schedule of more than one machines, then we choose a machine k and create two subproblems by (a) assigning job j solely to machine k and (b) the disallowing processing job j by machine k. These constraints can be enforced by removing the job j from the corresponding constraint sets $J_{k'}^i$.

2. If more than one job appears in the (fractional) schedule of a certain machine k at a position i, we chose one and create two subproblem by either (a) allowing this job only to be processed at positon k and (b) disallowing it to be processed at position k.

It is easy to see that if neither of the above branching rules are applicable for an *optimal* solution of the linear problem obtained by the column generation, then it is an optimal integer solution of the current subproblem.

In order to apply the early branching approach, we generate columns until the objective function value becomes lower than the best integer found so far. Then we continue generating, until either an

optimal solution is found or one of the branching rules becomes applicable. Then we choose the biggest non integer variable and branch according to that.

5 Conclusion

This paper presented an improved primal-dual method for solving the RMS problem of typical large scale combinatorial optimization problems which in turn allows implementing the B&B scheme with only partially solved subproblems. Our initial practical evaluation shows promising improvements on the reference applications compared to the existing solutions.

References

[1] R. Anbil, R. Tanga, E. L. Johnson, *A Global Approach to Crew-pairing Optimization*. IBM Systems J. 31, 71-78, 1992

[2] L. H. Appelgren, *A Column Generation Algorithms for a Ship Scheduling Problem*, Transportation Science 3, 53-68, 1969

[3] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W.P. Savelsbergh, P. H. Vance, *Branch-and-Price: Column Generation for Solving Huge Integer Programs*, Operations Research, 46 (1998), 316–329.

[4] J.-M. Bourjolly, G. Laporte, H. Mercure. *A combinatorial column generation algorithm for the maximum stable set problem.* Oper. Res. Lett. 20 (1997), 21–29.

[5] T. G. Crainic, J.-M. Rousseau, *The column generation principle and the airline crew pairing problem*, Infor, 25 (1987), 136–151.

[6] G. Desaulniers, J. Desrosiers, I. Ioachim, M. M. Solomon, F. Soumis, D. Villeneuve. *A unified framework for deterministic time con strained vehicle routing and crew scheduling problems.* In: T. G. Crainic, G. Laporte, eds. Fleet Management and Logistics. Kluwer, Norwell, MA, pp. 57-93, 1998.

[7] M. Desrochers, F. Soumis, *A Column Generation Approach to the Urban Transit Crew Scheduling Problem*. Transportation Science 23, 1-13, 1989.

[8] B. Dezső, A. Jüttner, P. Kovács, *Column Generation Method for an Agent Scheduling Problem*, Electronic Notes in Discrete Mathematics 36 (2010) 829836.

[9] O. du Merle, D. Villeneuve, J. Desrosiers, P. Hansen. *Stabilized column generation.* Discrete Math. 194 229237. 1999

[10] M. L. Fisher, J. S. Shapiro, Constructive Duality in Integer Programming, SIAM J. Appl. Math., 27 (1974), 31–52.

[11] M. L. Fisher, The Lagrangian Relaxation Method for Solving Integer Programming Problems, Management Science, 27 (1981), 1–18.

[12] L. R. Ford, D. R. Fulkerson, *A suggested computation for maximal multicommodity network flows.* Management Science, 5 (1958), 97–7101.

[13] A. M. Geoffrion, *Lagrangean relaxation for integer programming*, Mathematical Programming Study, 2 (1974), 82–114.

[14] M. E. Lübbecke, J. Desrosiers *Selected Topics in Column Generation*, Operations Research Vol. 53, No. 6, 2005, pp. 10071023

[15] R. E. Marsten, W. W. Hogan, J. W. Blankenship. *The Boxstep method for large-scale optimization.* Operations. Research. 23 389405. 1975

[16] A. Mehrotra, M. A. Trick, *A column generation approach for graph coloring.* INFORMS J. Comput, 8 (1996), 344–354.

[17] A Pessoa, R Sadykov, E Uchoa, F Vanderbeck, *Automation and combination of linear-programming based stabilization techniques in column generation* HAL, 2014, hal-01077984

[18] P. H. Vance, C. Barnhart, E. L. Johnson, G. L. Nemhauser, *Airline Crew Scheduling: A New Formulations and Decomposition Algorithm.* Operations Research 45, 188-200, 1997

[19] F. Vanderbeck, L. A. Wolsey, *Reformulation and Decomposition of Integer Programs*, In: M. Jünger et al. (eds.), 50 Years of Integer Programming 1958–2008, Springer-Verlag Berlin Heidelberg 2010, pp. 431–502.

[20] D. Villeneuve, J. Desrosiers, M. E. Lübbecke, *On Compact Formulations for Integer Programs Solved by Column Generation*, Annals of Operations Research 139 (2005), 375–388.

[21] P. Wentges. *Weighted dantzigwolfe decomposition for linear mixed-integer programming.* International Transactions in Operational Research, 4(2):151162, 1997

[22] W. E. Wilhelm, *A Technical Review of Column Generation in Integer Programming*, Optimization and Engineering, 2 (2001), 159–200.

[23] L. A. Wolsey, Integer programming duality, Mathematical Progamming, 20 (1981), 173–195.

Characterization of 1-Tough Graphs using Factors

Mikio Kano [1]

Ibaraki University
Hitachi, Ibaraki, Japan
mikio.kano.math@vc.ibaraki.ac.jp

Hongliang Lu [2]

School of Mathematics and Statistics
Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
luhongliang@mail.xjtu.edu.cn

Abstract: For a graph G, let $\omega(G)$ denote the number of components of G. In this paper we characterize a 1-tough graph G, which satisfies $\omega(G-S) \leq |S|$ for all $\emptyset \neq S \subset V(G)$, using an H-factor of a set-valued function $H : V(G) \to \{\{1\}, \{0,2,4,\ldots\}\}$.

Keywords: 1-factor, H-factor, 1-tough graph

1 Introduction

We consider finite simple graphs, which have neither loops nor multiple edges. We denote by $iso(G)$ and $odd(G)$ the number of isolated vertices and the number of odd components of G, respectively. For a set $\mathcal{S}$ of connected graphs, a spanning subgraph F of G is called an *$\mathcal{S}$-factor* if each component of F is isomorphic to an element of $\mathcal{S}$. For an integer $n \geq 3$, let C_n denote the cycle of order n, and K_2 denote the complete graph of order 2. Thus each component of a $\{K_2, C_n : n \geq 3\}$-factor of a graph is K_2 or a cycle, and a $\{K_2\}$-factor is nothing but a *1-factor*. A graph G is said to be *factor-critical* if for every vertex x of G, $G-x$ has a 1-factor. We begin with the 1-factor theorem.

Theorem 1 (The 1-factor theorem, [11]) *A connected graph G either has a 1-factor or is factor-critical if and only if*

$$odd(G-S) \leq |S| \qquad \text{for all} \quad \emptyset \neq S \subset V(G). \tag{1}$$

Assume that a connected graph G satisfies (1). If G has an even order, then G has a 1-factor, otherwise, G is factor critical. Moreover, the 1-factor theorem is usually stated as follows: a graph G has a 1-factor if and only if $odd(G-S) \leq |S|$ for all $S \subset V(G)$. By letting $S = \emptyset$ in this form, we obtain that every component of G is of even order. However as mentioned in the above theorem, if we use $\emptyset \neq S \subset V(G)$ instead of $S \subset V(G)$, then the order G is not necessary to be even, and if G has odd order and satisfies (1), then G is factor-critical. This fact is shown as follows:

It is known that a graph H of even order satisfies $odd(H-X) \equiv |X| \pmod 2$ for every $X \subset V(H)$. Assume that a connected graph G has odd order and satisfies (1), and let x be any vertex of G. Then $G-x$ is of even order, and for every $S \subset V(G-x)$, it follows from (1) and the property given above that

$$\begin{aligned} &odd(G-x-S) = odd(G-(S\cup\{x\})) \leq |S\cup\{x\}| = |S|+1 \quad \text{and} \\ &odd(G-x-S) \equiv |S| \pmod 2. \end{aligned}$$

Thus $odd(G-x-S) \leq |S|$. So $G-x$ has a 1-factor by the usual 1-factor theorem, and hence G is factor-critical. Conversely, if G is factor-critical, then for $\emptyset \neq S \subset V(G)$ and $y \in S$, we have $odd(G-S) = odd(G-y-(S-y)) \leq |S-y| \leq |S|$ since $G-y$ has a 1-factor. Hence (1) holds.

The next theorem is also well-known.

[1]Research is supported by JSPS KAKENHI Grant Number 16K05248

[2]Research is supported by he National Natural Science Foundation of China under grant No.11471257 and Fundamental Research Funds for the Central Universities

Theorem 2 ([12], Theorem 7.2 in [1]) *A connected graph G of order at least 2 has a $\{K_2, C_n : n \geq 3\}$-factor if and only if*

$$iso(G-S) \leq |S| \qquad \textit{for all} \quad \emptyset \neq S \subset V(G). \tag{2}$$

Since $iso(G-S) \leq odd(G-S)$, if a connected graph G of order at least 2 satisfies (1), then G satisfies (2), and so G has a $\{K_2, C_n : n \geq 3\}$-factor. This fact is explained as follows: Assume that G satisfies (1). If G has even order, then G has a 1-factor, which is a $\{K_2, C_n : n \geq 3\}$-factor. Assume that G has odd order, and let u and v be two adjacent vertices of G. Since G is factor-critical, $G-u$ has a 1-factor M_u and $G-v$ has a 1-factor M_v. Then $M_u \cup M_v$ is a union of two matchings of G, and each component of $M_u \cup M_v$ is a K_2, an even cycle or a path connecting u and v. Hence $(M_u \cup M_v) + uv$ is a $\{K_2, C_n : n \geq 3\}$-factor of G, which contains at most one odd cycle.

We denote by $\omega(G)$ the number of components of G. A connected graph G is said to be *t-tough* if $|S| \geq t\omega(G-S)$ for every $S \subset V(G)$ with $\omega(G-S) > 1$. It is obvious that

$$iso(G-S) \leq odd(G-S) \leq \omega(G-S) \qquad \text{for all} \quad \emptyset \neq S \subset V(G).$$

In this paper, we first characterize a connected graph G that satisfies $\omega(G-S) \leq |S|$ for all $\emptyset \neq S \subset V(G)$. Such a graph is called *1-tough*. Bauer, Hakimi and Schmeichel [3] showed that for any positive rational number t, the t-tough problem, which is a problem of checking a graph to be t-tough or nor, is NP-Hard. Later we generalize this characterization by using a function $f : V(G) \to \{1, 3, 5, \ldots\}$. Some results related to our theorems are found in [2, 4, 5, 6, 7, 9, 10].

2 Characterization of 1-tough graphs

We give a characterization of a graph G that satisfies $\omega(G-S) \leq |S|$ for all $\emptyset \neq S \subset V(G)$. In order to state our theorem, we need some notions and definitions. Let $\mathbf{Z}$ denote the set of integers. For two vertices x and y of a graph, an edge joining x to y is denoted by xy or yx. The degree of a vertex v in a subgraph H is denoted by $\deg_H(v)$. For two vertex sets X and Y of G, not necessary to be disjoint, we denote by $e_G(X, Y)$ the number of edges of G joining a vertex of X to a vertex of Y. If C is a component of $G-S$, then we briefly write $e_G(C, S)$ for $e_G(V(C), S)$. For a vertex set X of G, the subgraph of G induced by X is denoted by $\langle X \rangle_G$. For a function $h : V(G) \to \mathbf{Z}$, a subset $X \subseteq V(G)$ and a component C of $G-S$ for some $S \subset V(G)$, we write

$$h(X) := \sum_{x \in X} h(x) \quad \text{and} \quad h(C) := \sum_{x \in V(C)} h(x).$$

For any vertex x of G, let G^x denote the graph obtained from G by adding a new vertex x' together with a new edge xx', that is, $G^x = G + xx'$. Let $H : V(G) \to \{\{1\}, \{0, 2, 4, \ldots\}\}$ be a set-valued function. So $H(v)$ is equal to $\{1\}$ or $\{0, 2, 4, \ldots\}$ for each vertex v. We write

$$H^{-1}(1) := \{v \in V(G) : H(v) = \{1\}\}.$$

A spanning subgraph F of G is called an *H-factor* if $\deg_F(v) \in H(v)$ for all $v \in V(G)$. This H-factor is also called a *$\{1,even\}$-factor*. It is clear that if G has an H-factor, then $|H^{-1}(1)|$ must be even by the Handshaking Lemma. So if $|H^{-1}(1)|$ is odd, then G has no H-factor. For a function $H : V(G) \to \{\{1\}, \{0, 2, 4, \ldots\}\}$ and a vertex x of G, we define $H^x : V(G^x) \to \{\{1\}, \{0, 2, 4, \ldots\}\}$ as follows.

$$H^x(v) = \begin{cases} \{1\} & \text{if } v = x', \\ H(v) & \text{otherwise.} \end{cases} \tag{3}$$

A graph G is said to be *H-critical* or *$\{1,even\}$-critical* if G^x has an H^x-factor for every vertex x of G.

The next theorem is our first result, which gives a characterization of a 1-tough graph.

Theorem 3 *Let G be a connected graph. Then the following two statements hold.*

(i) *G has an H-factor for every $H : V(G) \to \{\{1\}, \{0,2,4,\ldots\}\}$ with $|H^{-1}(1)|$ even if and only if*

$$\omega(G-S) \leq |S|+1 \quad \textit{for all} \quad \emptyset \neq S \subset V(G). \tag{4}$$

(ii) *G is H-critical for every $H : V(G) \to \{\{1\}, \{0,2,4,\ldots\}\}$ with $|H^{-1}(1)|$ odd if and only if*

$$\omega(G-S) \leq |S| \quad \textit{for all} \quad \emptyset \neq S \subset V(G). \tag{5}$$

We generalize Theorem 3 by using an odd integer valued function f. Let G be a graph, and $f : V(G) \to \{1,3,5,\ldots\}$. Define a set-valued function H_f on $V(G)$ by

$$H_f(v) = \{1,3,\ldots,f(v)\} \quad \text{or} \quad \{0,2,4,\ldots\} \qquad \text{for each } v \in V(G). \tag{6}$$

Thus for a given function f, there are $2^{|V(G)|}$ set-valued functions H_f. For a set-valued function H_f on $V(G)$, define

$$H_f^{-1}(f) := \{v \in V(G) : H_f(v) = \{1,3,\ldots,f(v)\}\ \}.$$

A spanning subgraph F of G is called an *H_f-factor* if $\deg_F(v) \in H(v)$ for all $v \in V(G)$. This H_f-factor is also called an *$\{(1,f)$-odd,even$\}$-factor*. For a vertex x of G, we define a graph $G^x = G + xx'$. Moreover, for a function H_f on $V(G)$, define the function H_f^x on $V(G^x)$ as follows.

$$H_f^x(v) = \begin{cases} \{1\} & \text{if } v = x', \\ H_f(v) & \text{otherwise.} \end{cases}$$

A graph is said to be *H_f-critical* or *$\{(1,f)$-odd,even$\}$-critical* if G^x has an H_f^x-factor for every vertex x of G.

Theorem 4 *Let G be a connected graph, and let $f : V(G) \to \{1,3,5,\ldots\}$ be a function. Then the following two statements hold.*

(i) *G has an H_f-factor for every function H_f with $|H_f^{-1}(f)|$ even if and only if*

$$\omega(G-S) \leq f(S)+1 \quad \textit{for all} \quad \emptyset \neq S \subset V(G). \tag{7}$$

(ii) *G is H_f-critical for every function H_f with $|H_f^{-1}(f)|$ odd if and only if*

$$\omega(G-S) \leq f(S) \quad \textit{for all} \quad \emptyset \neq S \subset V(G). \tag{8}$$

We prove our theorem by using the following theorem, so called a *parity (g,f)-factor theorem*. Let $g, f : V(G) \to \mathbf{Z}$ be functions such that $g(v) \leq f(v)$ and $g(v) \equiv f(v) \pmod 2$ for all $v \in V(G)$, where we allow that $g(x) < 0$ and $\deg_G(y) < f(y)$ for some vertices x and y (see Theorem 6.1 in [1]). Then a spanning subgraph F of G is called a *parity (g,f)-factor* if

$$g(v) \leq \deg_F(v) \leq f(v) \quad \text{and} \quad \deg_F(v) \equiv f(v) \pmod 2$$

for all $v \in V(G)$. The following theorem gives a criterion for a graph to have a parity (g,f)-factor.

Theorem 5 (Lovász, [8], Theorem 6.1 in [1]) *Let G be a connected graph and $g, f : V(G) \to \mathbf{Z}$ such that $g(v) \leq f(v)$ and $g(v) \equiv f(v) \pmod 2$ for all $v \in V(G)$. Then G has a parity (g,f)-factor if and only if for any two disjoint subsets S, T of $V(G)$,*

$$\eta(S,T) = f(S) - g(T) + \sum_{x \in T} \deg_G(x) \ - e_G(S,T) - q(S,T) \geq 0, \tag{9}$$

where $q(S,T)$ denotes the number of components C of $G-S-T$, called q-odd components, such that $f(C) + e_G(C,T) \equiv 1 \pmod 2$.

Note that if (9) holds, then $\eta(\emptyset,\emptyset) = -q(\emptyset,\emptyset) \geq 0$, which implies that $|f(V(G))| \equiv 0 \pmod 2$. The following lemma is useful.

Lemma 6 *Let G, g, f, S, T and $\eta(S,T)$ be the same as Theorem 5. Then*

$$\eta(S,T) \equiv \sum_{x \in V(G)} f(x) \pmod 2.$$

References

[1] J. Akiyama and M. Kano, *Factors and Factorizations of Graphs*, *LNM* **1031** (Springer), (2011).

[2] A. Amahashi, On factors with all degree odd, *Graphs Combin.* **1** (1985), 111–114.

[3] D. Bauer, S.L. Hakimi, E. Schmeichel, Recognizing tough graphs is NP-hard, *Discrete Appl. Math.* **28** (1990), 191–195.

[4] G. Cornuéjols, General factors of graphs, *J. Combin. Theory Ser. B* **45** (1988), 185–198.

[5] Y. Cui and M. Kano, Some results on odd factors of graphs, *J. Graph Theory* **12** (1988), 327–333.

[6] Y. Egawa, M. Kano and Z. Yan, $(1,f)$-factors of graphs with odd property *Graphs and Combinatorics* **32** (2016), 103–110.

[7] H. Enomoto, B. Jackson, P. Katerinis, A. Saito, Toughness and the existence of k-factors, *J. Graph Theory* **9** (1985), 87–95.

[8] L. Lovász, The factorization of graphs. II, *Acta Math. Hungar* **23** (1972), 223–246.

[9] H. Lu, An Extension of Cui-Kano's Characterization Problem on Graph Factors, *J. Graph Theory* **81** (2016) 5-15.

[10] H. Lu and D.W.L. Wang, A Tutte-type characterization for graph factors, preprint.

[11] W.T. Tutte, The factorization of linear graphs, *J. London Math. Soc.* **22** (1947) 107–111.

[12] W.T. Tutte, The 1-factors of oriented graphs, *Proc. Amer. Math. Soc.* **4** (1953) 922–931.

Sufficient connectivity conditions for rigidity of symmetric frameworks

Viktória E. Kaszanitzky[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Magyar tudósok krt 2., Budapest, 1117, Hungary
kaszanitzky@cs.bme.hu

Bernd Schulze[2]

Department of Mathematics and Statistics,
Lancaster University,
Lancaster LA1 4YF, United Kingdom
b.schulze@lancaster.ac.uk

Abstract: It is a famous result of Lovász and Yemini that 6-connected graphs are rigid in the plane [5]. This was recently improved by Jackson and Jordán [3] who showed that 6-mixed connectivity is also sufficient for rigidity. Here we give sufficient connectivity conditions for both 'forced symmetric' and 'incidentally symmetric' infinitesimal rigidity in the plane.

Keywords: rigidity, symmetric frameworks, highly connected graphs

1 Introduction

A d-dimensional *(bar-joint) framework* is a pair $(\tilde{G},p)$, where $\tilde{G}=(\tilde{V},\tilde{E})$ is a finite simple graph and $p:\tilde{V}\to\mathbb{R}^d$ is a map. An *infinitesimal motion* of $(\tilde{G},p)$ is a function $u:\tilde{V}\to\mathbb{R}^d$ such that

$$\langle p_i-p_j,u_i-u_j\rangle=0 \quad \text{for all } \{i,j\}\in\tilde{E}, \tag{1}$$

where $u_i=u(i)$, $p_i=p(i)$ for each i. An infinitesimal motion u of $(\tilde{G},p)$ is a *trivial infinitesimal motion* if there exists a skew-symmetric matrix S and a vector t such that $u(i)=Sp(i)+t$ for all $i\in\tilde{V}$. $(\tilde{G},p)$ is *infinitesimally rigid* if every infinitesimal motion of $(\tilde{G},p)$ is trivial, and *infinitesimally flexible* otherwise.

A framework $(\tilde{G},p)$ is called *generic* if the coordinates of the image of p are algebraically independent over $\mathbb{Q}$. Laman's landmark result from 1970 gives a combinatorial characterisation of generic infinitesimally rigid frameworks in $\mathbb{R}^2$ [10]. In [5] Lovász and Yemini established sufficient graph connectivity conditions for the infinitesimal rigidity of generic frameworks in $\mathbb{R}^2$. Their result was recently improved by Jackson and Jordán in [3]. Analogous results for higher dimensions have not yet been found.

Since many structures in areas of application of rigidity theory exhibit non-trivial symmetries, the study of how symmetry impacts the rigidity and flexibility of frameworks has become a highly active research area in recent years [9]. There are two basic approaches to this problem. First, one may ask whether a framework is 'forced-symmetric rigid', i.e., whether it can only be deformed by breaking the original symmetry of the structure. Combinatorial characterisations of the graphs whose generic realisations (modulo the given symmetry constraints) are forced-symmetric rigid have been established for all symmetry groups in the plane, except for dihedral groups of order $2n$, where n is even [4, 6]. More generally, one may ask if a symmetric framework is infinitesimally rigid, i.e., wether it does not have *any* non-trivial deformations. This problem is more complex. However, combinatorial characterisations for symmetry-generic infinitesimal rigidity have recently been established for a number of cyclic groups in the plane [1, 7].

[1]Research is supported by EPSRC First Grant EP/M013642/1 and by the Hungarian Scientific Research Fund (OTKA, grant number K109240).

[2]Research is supported by EPSRC First Grant EP/M013642/1.

In this paper we extend the results in [3, 5] and establish sufficient connectivity conditions for symmetric frameworks to be symmetry-forced rigid, as well as infinitesimally rigid in the plane.

2 Rigidity of symmetric frameworks

2.1 Symmetric graphs

Let $\tilde{G} = (\tilde{V}, \tilde{E})$ be a finite simple graph. An *action* of a group Γ on $\tilde{G}$ is a group homomorphism $\theta : \Gamma \to \mathrm{Aut}(\tilde{G})$, where $\mathrm{Aut}(\tilde{G})$ denotes the automorphism group of $\tilde{G}$. An action θ is called *free* on $\tilde{V}$ (resp., $\tilde{E}$) if $\theta(\gamma)(i) \neq i$ for every $i \in \tilde{V}$ (resp., $\theta(\gamma)(e) \neq e$ for every $e \in \tilde{E}$) and every non-identity $\gamma \in \Gamma$. We say that a graph $\tilde{G}$ is Γ-*symmetric* (with respect to θ) if Γ acts on $\tilde{G}$ by θ. In the following we will frequently omit to specify the action θ if it is clear from the context. We then denote $\theta(\gamma)(i)$ by γi. For simplicity, we will assume throughout this paper that θ acts freely on $\tilde{V}$.

For a Γ-symmetric graph $\tilde{G} = (\tilde{V}, \tilde{E})$, the *quotient* Γ-*gain graph* of $\tilde{G}$ is the pair (G, ψ), where $G = (V, E)$ is the quotient graph of $\tilde{G}$, together with an orientation on the edges, and $\psi : E \to \Gamma$ is an edge labelling defined as follows. Each edge orbit Γe connecting Γi and Γj in $\tilde{G}/\Gamma$ can be written as $\{\{\gamma i, \gamma \circ \alpha j\} \mid \gamma \in \Gamma\}$ for a unique $\alpha \in \Gamma$. For each Γe, orient Γe from Γi to Γj in $\tilde{G}/\Gamma$ and assign to it the gain α. Then E is the resulting set of oriented edges, and ψ is the corresponding gain assignment. See Figure 1(b) for an example of a quotient Γ-gain graph.

Note that (G, ψ) is unique up to choices of representative vertices. Moreover, the orientation is only used as a reference orientation and may be changed, provided that we also modify ψ so that if e is an edge in one direction, and e^{-1} is the same edge in the opposite direction, then $\psi(e^{-1}) = \psi(e)^{-1}$.

Let $\tilde{G}$ be a finite simple Γ-symmetric graph and let (G, ψ) be its quotient Γ-gain graph. Then $\tilde{G}$ is called the *covering graph* of (G, ψ). Furthermore, the map $c : \tilde{G} \to G$ which maps every element of a vertex orbit of $\tilde{G}$ to its representative vertex in G and every element of an edge orbit of $\tilde{G}$ to its representative edge in G is called a *covering map*.

2.2 Symmetric frameworks

Let $\tilde{G}$ be a Γ-symmetric graph (with respect to $\theta : \Gamma \to \mathrm{Aut}(\tilde{G})$), and let Γ act on $\mathbb{R}^d$ via the homomorphism $\tau : \Gamma \to O(\mathbb{R}^d)$. A framework $(\tilde{G}, p)$ is called Γ-*symmetric* (with respect to θ and τ) if

$$\tau(\gamma)(p(i)) = p(\theta(\gamma)i) \qquad \text{for all } \gamma \in \Gamma \text{ and all } i \in \tilde{V}. \tag{2}$$

Let $G = (V, E)$ be the quotient Γ-gain graph of $\tilde{G}$ with the covering map $c : \tilde{G} \to G$. It is convenient to fix a representative vertex i of each vertex orbit $\Gamma i = \{\gamma i : \gamma \in \Gamma\}$, and define the *quotient* of p to be $p' : V \to \mathbb{R}^d$, so that there is a one-to-one correspondence between p and p' given by $p(i) = p'(c(i))$ for each representative vertex i.

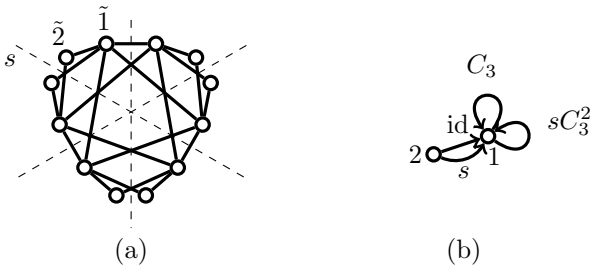


Figure 1: A framework with dihedral symmetry $\tau(\Gamma) = \mathcal{C}_{3v} = \langle s, C_3 \rangle$ (a) and its corresponding quotient Γ-gain graph (b).

For the group $\tau(\Gamma)$, let $\mathbb{Q}_\Gamma$ be the field generated by $\mathbb{Q}$ and the entries of the matrices in $\tau(\Gamma)$. We say that p is Γ-*generic* if the set of coordinates of the image of p' is algebraically independent over $\mathbb{Q}_\Gamma$. Note

that this definition does not depend on the choice of representative vertices. A Γ-symmetric framework $(\tilde{G}, p)$ is called Γ-*generic* if p is Γ-generic.

Throughout this paper, we will use the Schoenflies notation to describe the symmetries of frameworks. Note that in dimension 2, $\tau(\Gamma)$ can only be a reflection group of order 2 (denoted by $\mathcal{C}_s$), a rotational group of order n generated by a rotation C_n about the origin by $2\pi/n$, $n \in \mathbb{N}$ (denoted by $\mathcal{C}_n$), or a dihedral group of order $2n$ generated by a reflection and a rotation C_n (denoted by $\mathcal{C}_{nv}$).

2.3 Forced-symmetric rigidity

An infinitesimal motion u of a Γ-symmetric framework $(\tilde{G}, p)$ is called Γ-*symmetric* (with respect to θ and τ) if the velocity vectors exhibit the same symmetry as $(\tilde{G}, p)$, that is, if $\tau(\gamma)u_i = u_{\gamma i}$ for all $\gamma \in \Gamma$ and all $i \in \tilde{V}$. Moreover, we say that $(\tilde{G}, p)$ is Γ-*symmetric infinitesimally rigid* if every Γ-symmetric infinitesimal motion is trivial.

A key motivation for studying Γ-symmetric infinitesimal rigidity is that for Γ-generic frameworks, there exists a non-trivial Γ-symmetric infinitesimal motion if and only if there exists a non-trivial symmetry-preserving *continuous* motion [8, 9].

For a d-dimensional Γ-symmetric framework $(\tilde{G}, p)$, a symmetric analog of the rigidity matrix $R(\tilde{G}, p)$ [10], known as the *orbit rigidity matrix* was introduced in [8]. This matrix is of size $|E| \times d|V|$ and completely describes the Γ-symmetric infinitesimal rigidity properties of $(\tilde{G}, p)$. In particular, its kernel is isomorphic to the space of Γ-symmetric infinitesimal motions of $(\tilde{G}, p)$. We define the *rigidity matroid* of (G, ψ), $\mathcal{R}_\tau(G, \psi)$, to be the row matroid of the orbit rigidity matrix of a Γ-generic realisation of G (with respect to θ and τ). The bases of this matroid have been characterised for $\mathcal{C}_s$, $\mathcal{C}_n$, $n \in \mathbb{N}$, and $\mathcal{C}_{(2n+1)v}$, $n \in \mathbb{N}$, in [4, 6]. (For the groups $\mathcal{C}_{(2n)v}$, however, this problem is still open [4].) To state these results, we need the following definitions.

Let (G, ψ) be a quotient Γ-gain graph. The *gain* $\psi(W)$ of a closed walk W in (G, ψ) of the form $v_1, e_1, v_2, e_2, v_3, \ldots, v_k, e_k, v_1$ is defined as $\Pi_{i=1}^{k} \psi(e_i)^{\text{sign}(e_i)}$, where $\text{sign}(e_i) = 1$ if e_i is directed from v_i to v_{i+1}, and $\text{sign}(e_i) = -1$ otherwise. For $E' \subseteq E$ and $i \in V(E')$, we define the *subgroup of* Γ *induced by* E' as $\langle E' \rangle_{\psi,i} = \{\psi(W) : W \in \mathcal{W}(E', i)\}$, where $\mathcal{W}(E', i)$ is the set of closed walks starting at i using only edges of E'. A connected subset $E' \subseteq E$ is called *balanced* if $\langle E' \rangle_{\psi,i} = \{\text{id}\}$ for some $i \in V(E')$ (or equivalently for all $i \in V(E')$). Further, a connected subset $E' \subseteq E$ is called *cyclic* if $\langle E' \rangle_{\psi,i}$ is a cyclic subgroup of Γ for some $i \in V(E')$ (or equivalently for all $i \in V(E')$). A (possibly disconnected) subset $E' \subseteq E$ is called *balanced* (*cyclic*, resp.) if each of its connected components is balanced (cyclic). A subset $E' \subseteq E$ which is not balanced is called *unbalanced*.

Given a quotient Γ-gain graph (G, ψ), let ρ be the function on E defined by $\rho(X) = 2|V(X)| - 3 + \beta(X)$ for $X \subseteq E$ where

$$\beta(X) = \begin{cases} 0 \text{ if X is balanced;} \\ 2 \text{ if X is unbalanced and cyclic;} \\ 3 \text{ otherwise.} \end{cases}$$

Theorem 1 *[4] Let $(\tilde{G}, p)$ be a Γ-generic framework (with respect to θ and τ) such that $\tau(\Gamma)$ is $\mathcal{C}_s$, $\mathcal{C}_n$ or $\mathcal{C}_{(2n+1)v}$. Further, let (G, ψ) be the quotient Γ-gain graph of $\tilde{G}$ with $G = (V, E)$, and let $E' \subseteq E$. Then E' is independent in $\mathcal{R}_\tau(G, \psi)$ if and only if $\rho(F) \geq |F|$ for all $\emptyset \neq F \subseteq E'$. Further, $(\tilde{G}, p)$ is Γ-symmetric infinitesimally rigid if and only if (G, ψ) contains a spanning independent set of $2|V| - 1$ or $2|V|$ edges, depending on wether Γ is a non-trivial cyclic or dihedral group.*

The rank function of $\mathcal{R}_\tau(G, \psi)$ is given by the following formula.

Theorem 2 *[4] Let $(\tilde{G}, p)$ be a Γ-symmetric framework (with respect to θ and τ) such that $\tau(\Gamma)$ is $\mathcal{C}_s$, $\mathcal{C}_n$ or $\mathcal{C}_{(2n+1)v}$. Further let (G, ψ) be the quotient Γ-gain graph of $\tilde{G}$ with $G = (V, E)$. The rank of a set $E' \subseteq E$ in $\mathcal{R}_\tau(G, \psi)$ is equal to*

$$\min \left\{ \sum_{i=1}^{s} \rho(E_i) : \{E_1, \ldots, E_s\} \text{ is a partition of } E' \right\}.$$

2.4 Infinitesimal rigidity of symmetric frameworks

Combinatorial characterisations of Γ-generic infinitesimally rigid frameworks have been established for a selection of cyclic groups in [7]. We need the following definitions.

Let Γ be the group $\mathbb{Z}_k = \{0, 1, \ldots, k-1\}$ and for $t = 0, 1, \ldots, k-1$, let $\iota_t : \Gamma \to \mathbb{C} \setminus \{0\}$ be the irreducible representation of Γ defined by $\iota_t(j) = \omega^{tj}$, where ω denotes the root of unity $e^{\frac{2\pi i}{k}}$. For a Γ-symmetric framework $(\tilde{G}, p)$, an infinitesimal motion $u : \tilde{V} \to \mathbb{R}^d$ of $(\tilde{G}, p)$ is called *ι_t-symmetric* if it satisfies

$$\tau(\gamma)u_i = \omega^{t\gamma} u_{\gamma i} \qquad \text{for all } \gamma \in \Gamma \text{ and all } i \in \tilde{V}.$$

A Γ-symmetric framework $(\tilde{G}, p)$ is called *ι_t-symmetric infinitesimally rigid* if every ι_t-symmetric infinitesimal motion of $(\tilde{G}, p)$ is trivial.

Theorem 3 *[7] A Γ-generic framework $(\tilde{G}, p)$ (with respect to θ and τ) is infinitesimally rigid if and only if it is ι_t-symmetric infinitesimally rigid for every irreducible representation ι_t of Γ.*

Note that ι_0 is the trivial irreducible representation of Γ which assigns 1 to each $\gamma \in \Gamma$. Therefore, a framework is ι_0-symmetric infinitesimally rigid if and only if it is Γ-symmetric infinitesimally rigid.

For a Γ-symmetric framework $(\tilde{G}, p)$, an orbit rigidity matrix $O_t(\tilde{G}, p)$ was introduced in [7] for each irreducible representation ι_t of Γ. (For $t = 0$, the matrix $O_0(\tilde{G}, p)$ is the orbit rigidity matrix discussed in Section 2.3). Analogous to the case $t = 0$, the matrix $O_t(\tilde{G}, p)$ completely describes the ι_t-symmetric infinitesimal rigidity properties of $(\tilde{G}, p)$ for each t. We define the *ι_t-symmetric rigidity matroid* of (G, ψ), $\mathcal{R}^t_\tau(G, \psi)$, to be the row matroid of $O_t(\tilde{G}, p)$ for a Γ-generic realisation of G (with respect to θ and τ).

Given a quotient Γ-gain graph (G, ψ), where $\Gamma = \mathbb{Z}_2$, let μ be the function on E defined by $\mu(X) = 2|V(X)| - 3 + \beta_1(X)$ for $X \subseteq E$ where

$$\beta_1(X) = \begin{cases} 0 \text{ if X is balanced;} \\ 1 \text{ otherwise.} \end{cases}$$

Theorem 4 *[7] Let $(\tilde{G}, p)$ be a Γ-generic framework (with respect to θ and τ) such that $\tau(\Gamma)$ is $\mathcal{C}_s$ or $\mathcal{C}_2$. Further, let (G, ψ) be the quotient Γ-gain graph of $\tilde{G}$ with $G = (V, E)$, and let $E' \subseteq E$. Then E' is independent in $\mathcal{R}^1_\tau(G, \psi)$ if and only if $\mu(F) \geq |F|$ for all $\emptyset \neq F \subseteq E'$. Further, $(\tilde{G}, p)$ is ι_1-symmetric infinitesimally rigid if and only if (G, ψ) contains a spanning independent set of $2|V| - 2$ edges.*

By Theorem 3, Theorems 1 and 4 provide a combinatorial characterisation of Γ-generic infinitesimally rigid frameworks for $\Gamma = \mathbb{Z}_2$.

Note that the matroid $\mathcal{R}^1_\tau(G, \psi)$ is the Dilworth truncation of the union of the graphic matroid and the frame matroid (or bias matroid) of (G, ψ). We have the following formula for the rank function of $\mathcal{R}^1_\tau(G, \psi)$.

Theorem 5 *Let $(\tilde{G}, p)$ be a Γ-symmetric framework (with respect to θ and τ) such that $\tau(\Gamma)$ is $\mathcal{C}_s$ or $\mathcal{C}_2$. Further let (G, ψ) be the quotient Γ-gain graph of $\tilde{G}$ with $G = (V, E)$. The rank of a set $E' \subseteq E$ in $\mathcal{R}^1_\tau(G, \psi)$ is equal to*

$$\min \left\{ \sum_{i=1}^{s} \mu(E_i) \colon \{E_1, \ldots, E_s\} \text{ is a partition of } E' \right\}.$$

It was shown in [7] that for the three-fold rotational group $\tau(\Gamma) = \mathcal{C}_3$, a Γ-generic framework is Γ-symmetric infinitesimally rigid if and only if it is infinitesimally rigid.

The only other groups for which combinatorial characterisations of Γ-generic infinitesimally rigid frameworks have been found are the cyclic groups of order n, where $n < 1000$ is odd [1].

A *split* of vertex v of a quotient Γ-gain graph (G, ψ) is defined as follows. We can assume that every edge incident with v is directed from v. Take a 2-partition E_1, E_2 of non-loop edges incident with v. Replace v with a pair of vertices v_1, v_2. Replace every edge $vu \in E_i$ with edge $v_i u$ of the same label for

$i = 1, 2$. Then replace every (necessarily unbalanced) loop incident with v with an arc v_1v_2 of the same label. We say that a connected set F is *near-balanced* if it is not balanced and there is a split of (G, ψ) in which F results in a balanced set.

Given a quotient Γ-gain graph (G, ψ) and an irreducible representation ι_t of Γ, let ν_t be the function on E defined by $\nu_t(X) = 2|V(X)| - 3 + \alpha_t(X)$ for $X \subseteq E$ where

$$\alpha_t(X) = \begin{cases} 0 & \text{if X is balanced;} \\ 2 & \text{if X is near-balanced or satisfies } \langle X\rangle_{\psi,i} \simeq \mathbb{Z}_l \text{ for some} \\ & \quad l \in \{k' \in \mathbb{N} \mid 2 \le k' \le k;\ t \equiv 0 \text{ or } 1 \text{ or } -1 \pmod{k'}\}; \\ 3 & \text{otherwise.} \end{cases}$$

Theorem 6 *[2] Let $(\tilde{G}, p)$ be a Γ-generic framework (with respect to θ and τ) such that $\tau(\Gamma)$ is $\mathcal{C}_n$, $n \ge 5$ odd. Further, let (G, ψ) be the quotient Γ-gain graph of $\tilde{G}$ with $G = (V, E)$, and let $E' \subseteq E$.*

(i) Then E' is independent in $\mathcal{R}^t_\tau(G, \psi)$ if and only if $\nu_t(F) \ge |F|$ for all $\emptyset \ne F \subseteq E'$.

(ii) Further, $(\tilde{G}, p)$ is ι_t-symmetric infinitesimally rigid if and only if (G, ψ) contains a spanning independent set of $2|V| - 1$ edges if $t = 0, 1, k - 1$, and a spanning independent set of $2|V|$ edges otherwise.

(iii) The rank of E in the matroid is equal to

$$\min\left\{|E_0| + \sum_{i=1}^{s} \nu_t(E_i) \mid E_0 \subseteq E, E_1, \ldots, E_s \text{ are the edge sets of the connected components of } G - E_0\right\}.$$

3 Sufficient conditions for forced-symmetric rigidity

Following [3], we say that a graph $\tilde{G} = (\tilde{V}, \tilde{E})$ is *k-mixed-connected* if $\tilde{G} - U - D$ is connected for all sets $U \subseteq \tilde{V}$ and $D \subseteq \tilde{E}$ which satisfy $2|U| + |D| \le k - 1$. Note that for $k = 6$, for example, $\tilde{G}$ is 6-mixed-connected if and only if $\tilde{G}$ is 6-edge-connected, $\tilde{G} - v$ is 4-edge-connected and $\tilde{G} - v - u$ is 2-edge-connected for every $v, u \in \tilde{V}$. In this section we show the following main theorem:

Theorem 7 *Let $(\tilde{G}, p)$ be a Γ-generic framework (with respect to θ and τ) such that $\tau(\Gamma)$ is $\mathcal{C}_s$, $\mathcal{C}_n$ or $\mathcal{C}_{(2n+1)v}$, and let (G, ψ) be the quotient Γ-gain graph of $\tilde{G}$. Suppose $\tilde{G}$ is 6-mixed-connected. If $|\Gamma| \ge 6$ then suppose further that (G, ψ) is 2-edge-connected. Then $(\tilde{G}, p)$ is Γ-symmetric infinitesimally rigid.*

To prove this result, we first need the following definitions. For a graph $\tilde{G} = (\tilde{V}, \tilde{E})$ and disjoint sets $X, Y \subseteq \tilde{V}$, we let $d_{\tilde{G}}(X, Y)$ denote the number of edges between X and Y, and we let $d_{\tilde{G}}(X) := d_{\tilde{G}}(X, \tilde{V} \setminus X)$. In particular, $d_{\tilde{G}}(v) = d_{\tilde{G}}(\{v\})$ for $v \in \tilde{V}$.

A set $\mathcal{X} \subseteq 2^{\tilde{V}}$ is called a *cover* of $\tilde{G}$ if $\tilde{E} = \cup_{X \in \mathcal{X}} \tilde{E}(X)$. For a partition $\mathcal{P} = \{E_1, \ldots, E_s\}$ of the edge set E we define a cover $\mathcal{X}$ of $\tilde{G}$ as follows. Consider $E_i \in \mathcal{P}$ and let Γ_i be the subgroup induced by E_i. There exists a labelling ψ_i equivalent to ψ such that the label of every edge in E_i is an element of Γ_i [4]. Choose a representing element of every vertex orbit of $\tilde{G}$ such that the chosen elements define ψ_i. Let $\tilde{V}_i \subseteq \tilde{V}$ contain those of the representing elements which correspond to the vertex orbits of $V(E_i)$. Then the vertex set corresponding to E_i is $X_i = \Gamma_i\tilde{V}_i$. Every vertex set γX_i with $\gamma \in \Gamma$ belongs to $\mathcal{X}$. (Note that these sets are not necessarily pairwise distinct.) Thus every $E_i \in \mathcal{P}$ defines $|\Gamma|/|\Gamma_i|$ vertex sets in $\mathcal{X}$ and $\mathcal{X} = \{X \subseteq V : X = \gamma X_i \text{ for some } \gamma \in \Gamma, E_i \in \mathcal{P}\}$. We will call a cover of $\tilde{E}$ that can be obtained from a partition of E by applying the above process a *symmetric cover*.

We will use the following notation. For $X \in \mathcal{X}$ let $E_X = E_i$ for which there is a $\gamma \in \Gamma$ with $\gamma X_i = X$. Further, we let $\mathcal{X}_u = \{X \in \mathcal{X} : |\Gamma_X| \ge 4\}$ and $\mathcal{X}_3 = \{X \in \mathcal{X} : |X| \ge 3\}$.

The following is the key lemma to prove Theorem 7.

Lemma 8 *Suppose that $\tilde{G}$ is Γ-symmetric and 6-mixed-connected. If $|\Gamma| \geq 6$ then suppose further that (G, ψ) is 2-edge-connected. Then for every symmetric cover $\mathcal{X}$, we have*

$$\sum_{X \in \mathcal{X}} (2|X| - 3) \geq 2|\tilde{V}| + \sum_{X \in \mathcal{X}_3 \cap \mathcal{X}_u} (|\Gamma_X| - 3).$$

PROOF: Let $F = \bigcup_{X \in \mathcal{X}_3} \tilde{E}(X)$. With this notation, we have $\sum_{X \in \mathcal{X}} (2|X| - 3) = \sum_{X \in \mathcal{X}_3} (2|X| - 3) + |\tilde{E} - F|$.

Let $Y_X = X \bigcap \bigcup_{X' \in \mathcal{X}_3, X' \neq X} X'$ and $\mathcal{X}' = \{X \in \mathcal{X}_3 : X \neq Y_X\}$. Then

$$|\tilde{E} - F| \geq \frac{1}{2} \left(\sum_{X \in \mathcal{X}'} d_{\tilde{G} - Y_X}(X - Y_X) + \sum_{v \in \tilde{V} - \tilde{V}(\mathcal{X}_3)} d_{\tilde{G}}(v) \right).$$

By the 6-mixed-connectivity of $\tilde{G}$ we have $d_{\tilde{G} - Y_X}(X - Y_X) \geq \max\{6 - 2|Y_X|, 0\}$ for all $X \in \mathcal{X}'$. Suppose first that for $X \in \mathcal{X}$ we have $Y_X \neq \emptyset$. Observe that if $v \in Y_X$ for some $v \in \tilde{V}$, then $\gamma v \in Y_X$ for every $\gamma \in \Gamma_X$. Thus $|Y_X| \geq |\Gamma_X|$, and if $|\Gamma_X| \geq 4$, then $d_{\tilde{G} - Y_X}(X - Y_X) \geq 0 \geq 6 - 2|Y_X| + (2|\Gamma_X| - 6)$. If $Y_X = \emptyset$, then the same inequality holds, since $d_{\tilde{G}}(X) \geq 2|\Gamma_X|$ by the 2-edge-connectivity of (G, ψ). Thus

$$|\tilde{E} - F| \geq 3|\mathcal{X}'| - \sum_{X \in \mathcal{X}'} |Y_X| + \sum_{X \in \mathcal{X}' \cap \mathcal{X}_u} (|\Gamma_X| - 3) + 3(|\tilde{V}| - |\tilde{V}(\mathcal{X}_3)|).$$

Using this we have

$$\sum_{X \in \mathcal{X}_3} (2|X| - 3) + |\tilde{E} - F| \geq$$

$$\geq 2 \sum_{X \in \mathcal{X}_3} |X| - 3|\mathcal{X}_3| + 3|\mathcal{X}'| - \sum_{X \in \mathcal{X}'} |Y_X| + \sum_{X \in \mathcal{X}' \cap \mathcal{X}_u} (|\Gamma_X| - 3) + 3(|\tilde{V}| - |\tilde{V}(\mathcal{X}_3)|).$$

For every $X \in \mathcal{X}_3 - \mathcal{X}'$ we have $|Y_X| = |X| \geq \max\{3, |\Gamma_X|\}$. Thus

$$3|\mathcal{X}_3| - 3|\mathcal{X}'| = 3|\mathcal{X}_3 - \mathcal{X}'| \leq \sum_{X \in \mathcal{X}_3 - \mathcal{X}'} |Y_X| - \sum_{X \in (\mathcal{X}_3 - \mathcal{X}') \cap \mathcal{X}_u} (|\Gamma_X| - 3).$$

Using that $|\tilde{V}| - |\tilde{V}(\mathcal{X}_3)| \geq 0$ this implies

$$\sum_{X \in \mathcal{X}_3} (2|X| - 3) + |\tilde{E} - F| - 2|\tilde{V}| \geq$$

$$\geq 2 \sum_{X \in \mathcal{X}_3} |X| - \sum_{X \in \mathcal{X}_3 - \mathcal{X}'} |Y_X| - \sum_{X \in \mathcal{X}'} |Y_X| + \sum_{X \in \mathcal{X}_3 \cap \mathcal{X}_u} (|\Gamma_X| - 3) - 2|\tilde{V}(\mathcal{X}_3)| + (|\tilde{V}| - |\tilde{V}(\mathcal{X}_3)|)$$

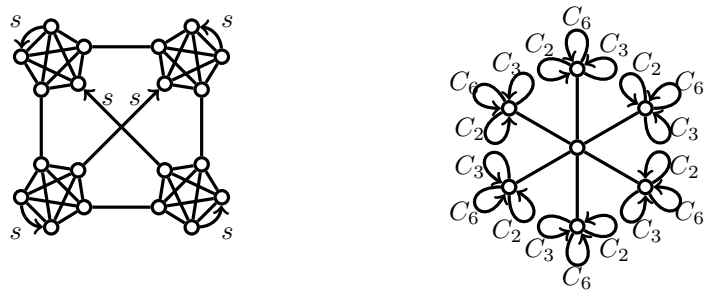
$$\geq 2 \sum_{X \in \mathcal{X}_3} (|X| - |Y_X|) + \sum_{X \in \mathcal{X}_3} |Y_X| + \sum_{X \in \mathcal{X}_3 \cap \mathcal{X}_u} (|\Gamma_X| - 3) - 2|\tilde{V}(\mathcal{X}_3)|.$$

$2 \sum_{X \in \mathcal{X}_3} (|X| - |Y_X|)$ is twice the number of vertices in $\tilde{V}(\mathcal{X}_3)$ contained by exactly one X. In $\sum_{X \in \mathcal{X}_3} |Y_X|$ every vertex contained in some Y_X with $X \in \mathcal{X}_3$ is counted at least twice. Thus $2 \sum_{X \in \mathcal{X}_3} (|X| - |Y_X|) + \sum_{X \in \mathcal{X}_3} |Y_X| \geq 2|\tilde{V}(\mathcal{X}_3)|$. Hence

$$\sum_{X \in \mathcal{X}} (2|X| - 3) = \sum_{X \in \mathcal{X}_3} (2|X| - 3) + |\tilde{E} - F| \geq 2|\tilde{V}| + \sum_{X \in \mathcal{X}_3 \cap \mathcal{X}_u} (|\Gamma_X| - 3)$$

as we claimed. □

We are now ready to prove Theorem 7.



(a) (b)

Figure 2: (a) An example of a $\mathbb{Z}_2$-gain graph (with $\mathbb{Z}_2 = \langle s \rangle$) whose covering graph is 5-mixed connected, but not $\mathbb{Z}_2$-symmetric infinitesimally rigid. (b) An example of a connected $\mathbb{Z}_6$-gain graph (with $\mathbb{Z}_6 = \langle C_6 \rangle$) whose covering graph is 6-mixed connected, but not $\mathbb{Z}_6$-symmetric infinitesimally rigid. In both (a) and (b), the orientation and edge label is omitted for every edge with gain id.

PROOF: Suppose for a contradiction that $\tilde{G}$ is 6-mixed-connected and the quotient Γ-gain graph $(G.\psi)$ is 2-edge-connected, but $(\tilde{G}, p)$ is not Γ-symmetric infinitesimally rigid. Equivalently, the edge set E of (G, ψ) has a partition $\mathcal{P} = \{E_1, \ldots, E_s\}$ with $\sum_{i=1}^{s} \rho(E_X) \leq 2|V| - 2$ if $\tilde{G}$ has rotational or reflectional symmetry or $\sum_{i=1}^{s} \rho(E_X) \leq 2|V| - 1$ if $\tilde{G}$ has dihedral symmetry.

Construct the symmetric cover $\mathcal{X}$ of $\tilde{G}$ from $\mathcal{P}$. By the construction of $\mathcal{X}$

$$|X| = |\Gamma_X||V(E_X)| \text{ for every } X \in \mathcal{X}, \tag{3}$$

from which we obtain

$$2|V(E_X)| - 1 \geq \frac{2|X| - 3}{|\Gamma_X|} \text{ if } 2 \leq |\Gamma_X| \leq 3, \tag{4}$$

$$2|V(E_X)| - 1 = \frac{(2|X| - 3) - (|\Gamma_X| - 3)}{|\Gamma_X|} \text{ if } |\Gamma_X| \geq 4 \text{ and } \Gamma_X \text{ is cyclic}, \tag{5}$$

$$2|V(E_X)| \geq \frac{2|X| - 3}{|\Gamma_X|} \text{ if } \Gamma_X \text{ is dihedral}. \tag{6}$$

Let $\mathcal{P}_b = \{E_i : \Gamma_i \text{ is balanced}\}$, $\mathcal{P}_c = \{E_i : \Gamma_i \text{ is cyclic but not balanced}\}$, and $\mathcal{P}_d = \{E_i : \Gamma_i \text{ is dihedral}\}$. Using the observations above we obtain

$$|\Gamma| \sum_{i=1}^{t} \rho(E_X) = |\Gamma| \left(\sum_{E_X \in \mathcal{P}_b} (2|V(E_X)| - 3) + \sum_{E_X \in \mathcal{P}_c} (2|V(E_X)| - 1) + \sum_{E_X \in \mathcal{P}_d} 2|V(E_X)| \right)$$

$$\geq |\Gamma| \sum_{E_X \in \mathcal{P}_b} (2|X| - 3) + \sum_{E_X \in \mathcal{P}_c} \frac{|\Gamma|}{|\Gamma_X|}(2|X| - 3) - |\Gamma| \sum_{E_X \in \mathcal{P}_c, |\Gamma_X| \geq 4} \frac{|\Gamma_X| - 3}{|\Gamma_X|} + \sum_{E_X \in \mathcal{P}_d} \frac{|\Gamma|}{|\Gamma_X|}(2|X| - 3)$$

$$= \sum_{X \in \mathcal{X}} (2|X| - 3) - |\Gamma| \sum_{E_X \in \mathcal{P}_c, |\Gamma_X| \geq 4} \frac{|\Gamma_X| - 3}{|\Gamma_X|}$$

$$\geq 2|\tilde{V}| + \sum_{X \in \mathcal{X}_3 \cap \mathcal{X}_u} (|\Gamma_X| - 3) - |\Gamma| \sum_{E_X \in \mathcal{P}_c, |\Gamma_X| \geq 4} \frac{|\Gamma_X| - 3}{|\Gamma_X|} \geq 2|\tilde{V}| = 2|\Gamma||V|,$$

where the last inequality follows from Lemma 8. This is a contradiction which completes the proof. □

The examples in Figure 2 show that Theorem 7 is best possible. Similar examples are easily constructed for the other groups mentioned in Theorem 7.

4 Sufficient conditions for infinitesimal rigidity

4.1 Reflection and two-fold rotational symmetry

For every $n \in \mathbb{N}$, it is easy to construct Γ-generic frameworks with reflection symmetry $\tau(\Gamma) = \mathcal{C}_s$ or half-turn symmetry $\tau(\Gamma) = \mathcal{C}_2$ whose underlying graphs are n-connected but that are not ι_1-symmetric infinitesimally rigid. Take, for example, a realisation of the complete graph K_n and its symmetric copy, and a matching between them, with all matching edges fixed by the non-trivial element $\gamma \in \Gamma$. Such a framework is not ι_1-symmetric infinitesimally rigid because a fixed edge in the covering graph $\tilde{G}$ corresponds to a loop in the quotient Γ-gain graph (G, ψ), and such a loop is dependent by Theorem 4.

In the following, we therefore only consider the edges of $\tilde{G}$ that are not fixed. Let $\tilde{E}_\ell$ denote the set of fixed edges in $\tilde{G}$ and let $\tilde{G}_\ell$ be $\tilde{G} - \tilde{E}_\ell$. Also, let $\mathcal{X}_2 = \{X \in \mathcal{X}, E_X \text{ is unbalanced}\}$.

We will show the following main theorem:

Theorem 9 *Let $(\tilde{G}, p)$ be a Γ-generic framework (with respect to θ and τ) such that $\tau(\Gamma)$ is $\mathcal{C}_s$ or $\mathcal{C}_2$. If $\tilde{G}_\ell$ is 7-mixed-connected, then $(\tilde{G}, p)$ is ι_1-symmetric infinitesimally rigid.*

We need the following key lemma.

Lemma 10 *Suppose that $\tilde{G}_\ell$ is 7-mixed-connected. Then for every symmetric cover $\mathcal{X}$, we have*

$$\sum_{X \in \mathcal{X}} (2|X| - 3) \geq 2|\tilde{V}| + |\mathcal{X}_2|.$$

Proof: Let $F = \bigcup_{X \in \mathcal{X}_3} \tilde{E}(X)$. With this notation, we have

$$\sum_{X \in \mathcal{X}} (2|X| - 3) = \sum_{X \in \mathcal{X}_3} (2|X| - 3) + |\tilde{E} - F|.$$

Let $Y_X = X \bigcap \bigcup_{X' \in \mathcal{X}_3, X' \neq X} X'$ and $\mathcal{X}' = \{X \in \mathcal{X}_3 : X \neq Y_X\}$. Then

$$|\tilde{E} - F| \geq \frac{1}{2} \left(\sum_{X \in \mathcal{X}'} d_{\tilde{G}_\ell - Y_X}(X - Y_X) + \sum_{v \in \tilde{V} - \tilde{V}(\mathcal{X}_3)} d_{\tilde{G}}(v) \right).$$

By the 7-mixed-connectivity of $\tilde{G}_\ell$ we have $d_{\tilde{G}_\ell - Y_X}(X - Y_X) \geq \max\{7 - 2|Y_X|, 0\}$ for all $X \in \mathcal{X}'$. Observe that if $X \in \mathcal{X}_2$ then $|Y_X|$ has to be even. Suppose first that for $X \in \mathcal{X}' \cap \mathcal{X}_2$ we have $|Y_X| \geq 4$. Then $d_{\tilde{G}_\ell - Y_X}(X - Y_X) \geq 6 - 2|Y_X| + 2$. If $|Y_X| = 2$ then $d_{\tilde{G}_\ell - Y_X}(X - Y_X) \geq 3 = 7 - 2|Y_X|$ but as $d_{\tilde{G}_\ell - Y_X}(X - Y_X)$ must be even, we can also deduce $d_{\tilde{G}_\ell - Y_X}(X - Y_X) \geq 6 - 2|Y_X| + 2$. If $Y_X = \emptyset$ then $d_{\tilde{G}_\ell - Y_X}(X - Y_X) \geq 7 = 7 - 2|Y_X|$ and again by the parity argument $d_{\tilde{G}_\ell - Y_X}(X - Y_X) \geq 6 - 2|Y_X| + 2$. Thus

$$|\tilde{E} - F| \geq 3|\mathcal{X}'| - \sum_{X \in \mathcal{X}'} |Y_X| + |\mathcal{X}' \cap \mathcal{X}_2| + 3(|\tilde{V}| - |\tilde{V}(\mathcal{X}_3)|).$$

Using this we have

$$\sum_{X \in \mathcal{X}_3} (2|X| - 3) + |\tilde{E} - F|$$

$$\geq 2 \sum_{X \in \mathcal{X}_3} |X| - 3|\mathcal{X}_3| + 3|\mathcal{X}'| - \sum_{X \in \mathcal{X}'} |Y_X| + |\mathcal{X}' \cap \mathcal{X}_2| + 3(|\tilde{V}| - |\tilde{V}(\mathcal{X}_3)|).$$

For every $X \in \mathcal{X}_2 - \mathcal{X}'$ we have $|Y_X| = |X| \geq 4$ and for every $X \in \mathcal{X}_3 - \mathcal{X}'$ we have $|Y_X| = |X| \geq 3$. Thus

$$3|\mathcal{X}_3| - 3|\mathcal{X}'| = 3|\mathcal{X}_3 - \mathcal{X}'| \leq \sum_{X \in \mathcal{X}_3 - \mathcal{X}'} |Y_X| - |\mathcal{X}_2 - \mathcal{X}'|.$$

Using $|\tilde{V}| - |\tilde{V}(\mathcal{X}_3)| \geq 0$ this implies

$$\sum_{X \in \mathcal{X}_3} (2|X| - 3) + |\tilde{E} - F| - 2|\tilde{V}|$$

$$\geq 2 \sum_{X \in \mathcal{X}_3} |X| - \sum_{X \in \mathcal{X}_3 - \mathcal{X}'} |Y_X| - \sum_{X \in \mathcal{X}'} |Y_X| + |\mathcal{X}_2| - 2|\tilde{V}(\mathcal{X}_3)| + (|\tilde{V}| - |\tilde{V}(\mathcal{X}_3)|)$$

$$\geq 2 \sum_{X \in \mathcal{X}_3} (|X| - |Y_X|) + \sum_{X \in \mathcal{X}_3} |Y_X| + |\mathcal{X}_2| - 2|\tilde{V}(\mathcal{X}_3)|.$$

$2\sum_{X \in \mathcal{X}_3}(|X| - |Y_X|)$ is twice the number of vertices in $\tilde{V}(\mathcal{X}_3)$ contained by exactly one X. In $\sum_{X \in \mathcal{X}_3} |Y_X|$ every vertex contained in some Y_X with $X \in \mathcal{X}_3$ is counted at least twice. Thus $2\sum_{X \in \mathcal{X}_3}(|X| - |Y_X|) + \sum_{X \in \mathcal{X}_3} |Y_X| \geq 2|\tilde{V}(\mathcal{X}_3)|$. Hence

$$\sum_{X \in \mathcal{X}} (2|X| - 3) = \sum_{X \in \mathcal{X}_3} (2|X| - 3) + |\tilde{E} - F| \geq 2|\tilde{V}| + |\mathcal{X}_2|,$$

as we claimed. □

We are now ready to prove Theorem 9.

PROOF: Suppose for a contradiction that $\tilde{G}_\ell$ is 7-mixed-connected but $(\tilde{G}, p)$ is not ι_1-symmetric infinitesimally rigid. Equivalently, the edge set E of the quotient Γ-gain graph (G, ψ) has a partition $\mathcal{P} = \{E_1, \ldots, E_s\}$ with $\sum_{i=1}^{s} \mu(E_X) \leq 2|V| - 3$. Construct the symmetric cover $\mathcal{X}$ of $\tilde{G}$ from $\mathcal{P}$. We have

$$2\sum_{i=1}^{s} \mu(E_X) = 2\left(\sum_{E_X \in \mathcal{P}_b} (2|V(E_X)| - 3) + \sum_{E_X \in \mathcal{P}_c} (2|V(E_X)| - 2) \right)$$

$$= 2 \sum_{E_X \in \mathcal{P}_b} (2|X| - 3) + \sum_{E_X \in \mathcal{P}_c} (2|X| - 4) = \sum_{X \in \mathcal{X}} (2|X| - 3) - |\mathcal{X}_2| \geq 2|\tilde{V}| = 4|V|,$$

where the last inequality follows from Lemma 10. This is a contradiction which completes the proof. □

The example in Figure 3 shows that this result is best possible.

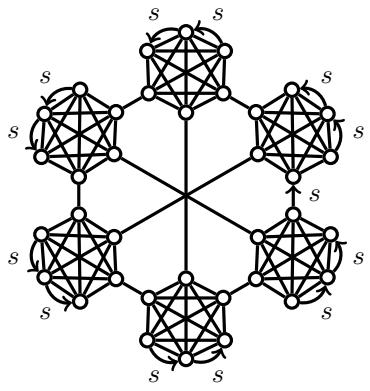


Figure 3: A $\mathbb{Z}_2$-gain graph (with $\mathbb{Z}_2 = \langle s \rangle$) whose covering graph is 6-mixed connected, but not ι_1-symmetric infinitesimally rigid. The orientation and edge label is omitted for every edge with gain id.

By Theorem 3, we may combine Theorem 7 and Theorem 9 to obtain the following.

Corollary 11 *Let $(\tilde{G}, p)$ be a Γ-generic framework (with respect to θ and τ) such that $\tau(\Gamma)$ is $\mathcal{C}_s$ or $\mathcal{C}_2$. If $\tilde{G}_\ell$ is 7-mixed-connected, then $(\tilde{G}, p)$ is infinitesimally rigid.*

4.2 Rotational symmetry of order $n \geq 3$

It was shown in [7] that a generic $\mathcal{C}_3$-symmetric framework is $\mathcal{C}_3$-symmetric infinitesimally rigid if and only if it is infinitesimally rigid. If we combine this with Theorem 7 we get the following sufficient condition:

Corollary 12 *Let $(\tilde{G}, p)$ be a $\mathcal{C}_3$-generic framework (with respect to θ and τ). If $\tilde{G}$ is 6-mixed-connected, then $(\tilde{G}, p)$ is infinitesimally rigid.*

For rotational groups $\mathcal{C}_n$ for odd n and $n \geq 5$ we can prove the following similar result:

Theorem 13 *Let $(\tilde{G}, p)$ be a $\mathcal{C}_n$-generic framework (with respect to θ and τ) where $n \geq 5$ and is odd, and let (G, ψ) be the quotient Γ-gain graph of $\tilde{G}$. Suppose $\tilde{G}_\ell$ is 6-mixed-connected. If $n \geq 7$ suppose further that G is 2-edge-connected. Then $(\tilde{G}, p)$ is infinitesimally rigid.*

Proof: Take an arbitrary cover of $E(G_\ell)$ in the form $E_0, E_1, \ldots, E_s$ where $E_0 \subseteq E$ and $E_1, \ldots, E_s$ are the edge sets of the connected components of $G - E_0$. By Theorem 6, G is infinitesimally rigid if and only if $|E_0| + \sum_{i=1}^{s} \nu_t(E_i) \geq 2|V| - 1$ for $t = 0, 1, n-1$ and otherwise $|E_0| + \sum_{i=1}^{s} \nu_t(E_i) \geq 2|V|$ holds for every $0 \leq t \leq n-1$.

We will give a lower bound for $|E_0| + \sum_{i=1}^{s} \nu_t(E_i)$. As every edge in $E(G_\ell)$ forms a balanced set, we have $\nu_t(\{e\}) = 1$ for every $e \in E(G_\ell)$. Thus

$$|E_0| + \sum_{i=1}^{s} \nu_t(E_i) = \sum_{e \in E_0} \nu_t(\{e\}) + \sum_{i=1}^{s} \nu_t(E_i) \geq \sum_{e \in E_0} \rho(\{e\}) + \sum_{i=1}^{s} \rho(E_i).$$

As $\{e : e \in E_0\}, E_1, \ldots, E_s$ is a cover of $E(G_\ell)$, by (the end of the proof of) Theorem 7 we get that $|E_0| + \sum_{i=1}^{s} \nu_t(E_i) \geq 2|V|$ which completes the proof. □

References

[1] R. Ikeshita, Infinitesimal rigidity of symmetric frameworks, Master Thesis, University of Tokyo, 2015.

[2] R. Ikeshita, S. Tanigawa, Count matroids of group-labeled graphs, *arXiv:1507.01259*, 2015.

[3] B. Jackson and T. Jordán, A sufficient connectivity condition for generic rigidity in the plane, *Discrete Applied Mathematics* **157**(8) (2009), 1965–1968.

[4] T. Jordán, V.E. Kaszanitzky and S. Tanigawa, Gain-sparsity and symmetry-forced rigidity in the plane, *Discrete Comput. Geom.* **55**(2) (2016), 314–372.

[5] L. Lovász and Y. Yemini, On generic rigidity in the plane, *SIAM J. Algebraic Discrete Methods* **3**(1) (1982), 91–98.

[6] J. Malestein and L. Theran, Frameworks with forced symmetry I: reflections and rotations, *Discrete Comput. Geom.* **54**(2) (2015), 339–367.

[7] B. Schulze and S. Tanigawa, Infinitesimal Rigidity of Symmetric Bar-Joint Frameworks, *SIAM J. Discrete Math.* **29**(3) (2015), 1259–1286.

[8] B. Schulze and W. Whiteley, The orbit rigidity matrix of a symmetric framework, *Discrete Comput. Geom.*, **46**(3) (2011), 561–598.

[9] B. Schulze and W. Whiteley, Rigidity and Symmetry, in *Handbook of Discrete and Computational Geometry*, Third Edition, Editors: Csaba D. Toth, Joseph O'Rourke, Jacob E. Goodman, Chapman and Hall/CRC, to appear in 2017.

[10] W. Whiteley, Some Matroids from Discrete Applied Geometry, *Contemporary Mathematics, AMS* **197** (1996), 171–311.

A general 2-part Erdős-Ko-Rado theorem

Gyula O.H. Katona[1]

MTA Rényi Institute
Budapest, Reáltanoda u. 13-15, Hungary
ohkatona@renyi.hu

Abstract: A two-part extension of the famous Erdős-Ko-Rado Theorem is proved. The underlying set is partitioned into X_1 and X_2. Some positive integers $k_i, \ell_i (1 \leq i \leq m)$ are given. We prove that if $\mathcal{F}$ is an intersecting family containing members F such that $|F \cap X_1| = k_i, |F \cap X_2| = \ell_i$ holds for one of the values $i (1 \leq i \leq m)$, $|X_1|$ and $|X_2|$ are large enough, then $|\mathcal{F}|$ cannot exceed the size of the largest subfamily containing one element.

Keywords: extremal set theory, two-part families, cyclic permutation

1 Introduction

Let X be a finite set of n elements. A family $\mathcal{F} \subset 2^X$ is called *intersecting* if $F, G \in \mathcal{F}$ implies $F \cap G \neq \emptyset$. The family of all k-element subsets of X is denoted by $\binom{X}{k}$. The celebrated theorem of Erdős, Ko and Rado is the following.

Theorem 1 [1] *Suppose that an integer $k \leq \frac{n}{2}$ is given and $\mathcal{F} \subset \binom{X}{k}$ is intersecting. Then*

$$|\mathcal{F}| \leq \binom{n-1}{k-1}.$$

The family of all k-element subsets containing a fixed element $x \in X$ shows that the estimate is sharp.

The goal of our paper is to consider the problem when the underlying set is partitioned into two parts X_1, X_2 and the sets $F \in \mathcal{F}$ have fixed sizes in both parts. More precisely let X_1 and X_2 be disjoint sets of n_1, respectively n_2 elements. [2] considered such subsets of $X = X_1 \cup X_2$ which had k elements in X_1 and ℓ elements in X_2. The family of all such sets is denoted by $\binom{X_1, X_2}{k, \ell}$. The construction above, taking all possible sets containing a fixed element also works here. If the fixed element is in X_1 then the number of these sets is

$$\binom{n_1 - 1}{k-1}\binom{n_2}{\ell},$$

otherwise it is

$$\binom{n_1}{k}\binom{n_2 - 1}{\ell - 1}.$$

The following theorem of Frankl [2] claims that the larger one of these is the best.

Theorem 2 *Let X_1, X_2 be two disjoint sets of n_1 and n_2 elements, respectively. The positive integers k, ℓ satisfy the inequalities $2k \leq n_1, 2\ell \leq n_2$. If $\mathcal{F}$ is an intersecting subfamily of $\binom{X_1, X_2}{k, \ell}$ then*

$$|\mathcal{F}| \leq \max\left\{\binom{n_1 - 1}{k-1}\binom{n_2}{\ell}, \binom{n_1}{k}\binom{n_2 - 1}{\ell - 1}\right\}.$$

[1]This research was supported by the National Research, Development and Innovation Office – NKFIH Fund No's 104183, SSN117879 and K116769.

2 Main result

The goal of the present paper is to generalize Theorem 2 for the case when other sizes are also allowed that is the family consists of sets satisfying $|F \cap X_1| = k_i, |F \cap X_2| = \ell_i$ for certain pairs of integers. Using the notation above, we will consider subfamilies of

$$\bigcup_{i=1}^{m} \binom{X_1, X_2}{k_i, \ell_i}.$$

The generalization is however a little weaker at one point. In Theorem 2 the thresholds $2k \leq n_1, 2\ell \leq n_2$ for validity are natural. If either n_1 or n_2 is smaller then the problem becomes trivial, all such sets can be selected in $\mathcal{F}$. In the generalization below there is no such natural threshold. There will be another difference in the formulation. We give the construction of the extremal family rather than the maximum number of sets. A family is called *trivially intersecting* if there is an element contained in every member.

Theorem 3 *Let X_1, X_2 be two disjoint sets of n_1 and n_2 elements, respectively. Some positive integers $k_i, \ell_i (1 \leq i \leq m)$ are given. Define $b = \max_i\{k_i, \ell_i\}$. Suppose that $9b^2 \leq n_1, n_2$. If $\mathcal{F}$ is an intersecting subfamily of*

$$\bigcup_{i=1}^{m} \binom{X_1, X_2}{k_i, \ell_i}$$

then $|\mathcal{F}|$ cannot exceed the size of the largest trivially intersecting family satisfying the conditions.

Sketch of the proof. (The full proof can be found in [4]

We will use the method of cyclic permutations [3] giving a simple proof of the EKR theorem. There the analogous problem is solved for intervals along a cyclic permutation and then a double counting easily finishes the proof. Here we need a pair of cyclic permutations: one for X_1 and one for X_2. A cycle of size n_i will be represented by the integers $\mod n_i$. The usual notation is $\mathbb{Z}_{n_i}$. Hence the pair of cycles will be $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$. The direct product of the intervals of length k and ℓ, in $\mathbb{Z}_{n_1}$ and $\mathbb{Z}_{n_2}$, respectively, will be a $k \times \ell$ *rectangle* in $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$. Problems analogous to our Theorem 3 will be considered for such rectangles.

Let $\mathcal{R}_i$ be a family of $k_i \times \ell_i$ rectangles in $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} (1 \leq i \leq m)$. We say that $\mathcal{R} = \bigcup_{i=1}^{m} \mathcal{R}_i$ is a *proj-intersecting family* if, for any two members either the projections on $\mathbb{Z}_{n_1}$ or on $\mathbb{Z}_{n_2}$ are intersecting. One can prove the statement analogous to Theorem 3 for the rectangles, that is, the largest $\mathcal{R}$ is trivially intersecting either in the projections in $\mathbb{Z}_{n_1}$ or in the projections in $\mathbb{Z}_{n_2}$. In other words

$$\sum_{i=1}^{m} |\mathcal{R}_i| \leq \max\left\{ n_1 \sum_{i=1}^{m} \ell_i, n_2 \sum_{i=1}^{m} k_i \right\}$$

holds. However this is not sufficient for the proof of the theorem. A weighted version is needed.

Lemma 4 *Suppose that the positive integers k_i, ℓ_i, b, n_1, n_2 satisfy the inequalities $k_i, \ell_i \leq b (1 \leq i \leq m), 9b^2 < n_1, n_2$. Let $\mathcal{R}_i$ be a family of $k_i \times \ell_i$ rectangles in $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} (1 \leq i \leq m)$. Suppose that $\mathcal{R} = \bigcup_{i=1}^{m} \mathcal{R}_i$ is a proj-intersecting family. Let $\lambda_i > 0 (1 \leq i \leq m)$ be real numbers. Then*

$$\sum_{i=1}^{m} \lambda_i |\mathcal{R}_i| \leq \max\left\{ n_1 \sum_{i=1}^{m} \lambda_i \ell_i, n_2 \sum_{i=1}^{m} \lambda_i k_i \right\}$$

holds.

Define the families

$$\mathcal{F}_i = \{F \in \mathcal{F} : \ |F \cap X_1| = k_i, |F \cap X_2| = \ell_i\}.$$

We use double counting for the sum

$$\sum_{F,\mathcal{C}_1,\mathcal{C}_2} s(F)$$

where $\mathcal{C}_j$ is a cyclic permutation of $\mathbb{Z}_{n_j} (j = 1, 2)$, $F \in \mathcal{F}$ and it forms a rectangle for the product of these two cyclic permutations and the weight $s(F)$ is defined in the following way:

$$s(F) = s_i(F) = \frac{1}{n_1!} \cdot \frac{1}{n_2!} \binom{n_1}{k_i} \binom{n_2}{\ell_i} \text{ if } F \in \mathcal{F}_i.$$

Some tedious calculations and the usage of Lemma 1 leads to the proof of Theorem 3.

References

[1] P. Erdős, Chao Ko, R. Rado, Intersection theorems for systems of finite sets, *Quarterly J. of Math. (Oxford)* **12** (1961) 313 – 320.

[2] P. Frankl, An Erdős Ko Rado Theorem for Direct Products, *Europ . J . Combinatorics* **17** (1996) 727 - 730.

[3] G.O.H. Katona, A simple proof of the Erdős-Chao Ko-Rado theorem, *J. Combin. Theory Ser B* **13** (1972) 183 – 184.

[4] G.O.H. Katona, A general 2-part Erdős-Ko-Rado theorem, arXiv:1703.00287.

The complexity of recognizing minimally tough graphs

Gyula Y Katona[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
1111 Budapest, Műegyetem rkpt. 3, Hungary
and
MTA-ELTE Numerical Analysis and Large Networks Research Group
kiskat@cs.bme.hu

István Kovács

Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
1111 Budapest, Műegyetem rkpt. 3, Hungary
kovika@iit.bme.hu

Kitti Varga[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
1111 Budapest, Műegyetem rkpt. 3, Hungary
vkitti@cs.bme.hu

Abstract: Let t be a positive real number. A graph is called t-tough, if the removal of any cutset S leaves at most $|S|/t$ components. The toughness of a graph is the largest t for which the graph is t-tough. A graph is minimally t-tough, if the toughness of the graph is t and the deletion of any edge from the graph decreases the toughness. The complexity class DP is the set of all languages that can be expressed as the intersection of a language in NP and a language in coNP. We prove that recognizing minimally t-tough graphs is DP-complete for any positive integer t and for any positive rational number $t \leq 1/2$.

Keywords: 3–6 keywords toughness, complexity, DP-complete

1 Introduction

All graphs considered in this paper are finite, simple and undirected. Let $\omega(G)$ denote the number of components and $\alpha(G)$ denote the independence number. For a graph G and a vertex set $V \subseteq V(G)$, let $G[V]$ denote the subgraph of G induced by V.

The complexity class DP was introduced by C. H. Papadimitriou and M. Yannakakis [4].

Definition 1 *A language L is in the class* DP *if there exist two languages $L_1 \in NP$ and $L_2 \in coNP$ such that $L = L_1 \cap L_2$.*

We mention that DP $\neq$ NP $\cap$ coNP, if NP $\neq$ coNP. Moreover, NP $\cup$ coNP $\subseteq$ DP. A language is called DP-hard if all problems in DP can be reduced to it in polynomial time. A language is DP-complete if it is in DP and it is DP-hard.

[1]Research is supported by by the National Research, Development and Innovation Office NKFIH (grant number K108947)

A critical-type DP-complete problem is CriticalClique [5], in our proofs we use an equivalent form of it, α-Critical.

CriticalClique
Instance: a graph G and a positive integer k.
Question: is it true that G has no clique of size k, but adding any missing edge e to G, the resulting graph $G + e$ has a clique of size k?

By taking the complement of the graph, we can obtain α-Critical from CriticalClique.

Definition 2 *A graph G is called* α-critical, *if $\alpha(G - e) > \alpha(G)$ for all $e \in E(G)$.*

α-Critical
Instance: a graph G and a positive integer k.
Question: is it true that $\alpha(G) < k$, but $\alpha(G - e) \geq k$ for any edge $e \in E(G)$?

Since a graph is clique-critical if and only if its complement is α-critical, α-Critical is also DP-complete.

Corollary 3 *α-Critical is DP-complete.*

The notion of toughness was introduced by Chvátal [2].

Definition 4 *Let t be a positive real number. A graph G is called* t-tough, *if*

$$\omega(G - S) \leq \frac{|S|}{t}$$

for any cutset S of G (i.e. for any S with $\omega(G - S) > 1$). The toughness *of G, denoted by $\tau(G)$, is the largest t for which G is t-tough, taking $\tau(K_n) = \infty$ for all $n \geq 1$.*
We say that a cutset $S \subseteq V(G)$ is a tough set *if $\omega(G - S) = |S|/\tau(G)$.*

For all positive rational number t we can define a separate problem:

***t*-Tough**
Instance: a graph G,
Question: is it true that $\tau(G) \geq t$?

Bauer et al. proved the following.

Theorem 5 ([1]) *For any positive rational number t, t-Tough is coNP-complete.*

The critical form of this problem is minimally toughness.

Definition 6 *A graph G is* minimally t-tough, *if $\tau(G) = t$ and $\tau(G - e) < t$ for all $e \in E(G)$.*

Given t we define:

Min-*t*-Tough
Instance: a graph G,
Question: is it true that G is minimally t-tough?

Our main result is the following.

Theorem 7 Min-t-Tough *is DP-complete for any positive integer t and for any positive rational number $t \leq 1/2$.*

First we prove this theorem for $t = 1$, then we generalize that proof for positive integers, and finally we prove it for any positive rational number $t \leq 1/2$.

2 Preliminaries

In this section we prove some useful lemmas.

Proposition 8 *Let G be a connected noncomplete graph on n vertices. Then $\tau(G) \in \mathbb{Q}^+$, and if $\tau(G) = a/b$, where a, b are positive integers and $(a, b) = 1$, then $1 \leq a, b \leq n - 1$.*

PROOF: By definition,

$$\tau(G) = \min_{\substack{S \subseteq V(G) \\ \text{cutset}}} \frac{|S|}{\omega(G - S)}$$

for a noncomplete graph G. Since G is connected and noncomplete, $1 \leq |S| \leq n - 2$ and since S is a cutset, $2 \leq \omega(G - S) \leq n - 1$. □

Corollary 9 *Let G and H be two connected noncomplete graphs on n vertices. If $\tau(G) \neq \tau(H)$, then*

$$|\tau(G) - \tau(H)| > \frac{1}{n^2}.$$

Claim 10 *For every positive rational number t,* MIN-t-TOUGH $\in$ *DP.*

PROOF: For any positive rational number t,

$$\begin{aligned} \text{MIN-}t\text{-TOUGH} &= \{G \text{ graph} \mid \tau(G) = t \text{ and } \tau(G - e) < t \text{ for all } e \in E(G)\} = \\ &= \{G \text{ graph} \mid \tau(G) \geq t\} \cap \{G \text{ graph} \mid \tau(G) \leq t\} \cap \\ &\quad \cap \{G \text{ graph} \mid \tau(G - e) < t \text{ for all } e \in E(G)\}. \end{aligned}$$

Let

$$L_{1,1} = \{G \text{ graph} \mid \tau(G - e) < t \text{ for all } e \in E(G)\},$$

$$L_{1,2} = \{G \text{ graph} \mid \tau(G) \leq t\}$$

and

$$L_2 = \{G \text{ graph} \mid \tau(G) \geq t\}.$$

$L_2 \in$ coNP, a witness is a cutset $S \subseteq V(G)$ whose removal leaves more than $|S|/t$ components. $L_{1,1} \in$ NP, the witness is a set of cutsets: $S_e \subseteq V(G)$ for each edge e whose removal leaves more than $|S_e|/t$ components.

Now we show that $L_{1,2} \in$ NP, i.e. we can express $L_{1,2}$ in a form of

$$L_{1,2} = \{G \text{ graph} \mid \tau(G) < t + \varepsilon\},$$

which belongs to NP. Let a, b be positive integers such that $t = a/b$ and $(a, b) = 1$, and let G be an arbitrary graph on n vertices. If G is disconnected, then $\tau(G) = 0$, and if G is complete, then $\tau(G) = \infty$, so in both cases G is not minimally t-tough. By Proposition 8, if $1 \leq a, b \leq n - 1$ does not hold, then G is also not minimally t-tough. So we can assume that $t = a/b$, where a, b are positive integers, $(a, b) = 1$ and $1 \leq a, b \leq n - 1$. With this assumption

$$L_{1,2} = \{G \text{ graph} \mid \tau(G) \leq t\} = \left\{ G \text{ graph} \;\middle|\; \tau(G) < t + \frac{1}{|V(G)|^2} \right\},$$

so $L_{1,2} \in$ NP.

Since $L_{1,1} \cap L_{1,2} \in$ NP, $L_2 \in$ coNP and MIN-t-TOUGH $= (L_{1,1} \cap L_{1,2}) \cap L_2$, we can conclude that MIN-t-TOUGH $\in$ DP. □

Claim 11 *Let t be a positive rational number and G a minimally t-tough graph. For every edge e of G,*

1. *the edge e is a bridge in G, or*
2. *there exists a vertex set $S = S(e) \subseteq V(G)$ with*

$$\omega(G - S) \leq \frac{|S|}{t} \quad \text{and} \quad \omega\big((G - e) - S\big) > \frac{|S|}{t},$$

and the edge e is a bridge in $G - S$.

In the first case, we define $S = S(e) = \emptyset$.

PROOF: Let e be an arbitrary edge of G, which is not a bridge. Since G is minimally t-tough, $\tau(G - e) < t$. So there exists a cutset $S = S(e) \subseteq V(G - e) = V(G)$ in $G - e$ satisfying $\omega\big((G - e) - S\big) > |S|/t$. On the other hand, $\tau(G) = t$, so $\omega(G - S) \leq |S|/t$. This is only possible if e connects two components of $(G - e) - S$. $\square$

Finally we cite a Lemma that our proof relies on.

Lemma 12 (Problem 14 **of 8 in** [3]**)** *If we replace a vertex of an α-critical graph with a clique, and connect every neighbor of the original vertex with every vertex in the clique, then the resulting graph is still α-critical.*

3 Recognizing minimally 1-tough graphs

To show that MIN-1-TOUGH is DP-hard, we reduce α-CRITICAL to it.

Theorem 13 MIN-1-TOUGH *is DP-complete.*

PROOF: In Claim 10 we have already proved that MIN-1-TOUGH $\in$ DP.

Let G be an arbitrary connected graph on the vertices $v_1, \ldots, v_n$. Let G_α be defined as follows. It will be easy to see that it can be constructed from G in polynomial time. For all $i \in [n]$, let

$$V_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,\alpha}\}$$

and place a clique on the vertices of V_i. For all $i, j \in [n]$, if $v_i v_j \in E(G)$, then place a complete bipartite graph on $(V_i; V_j)$. For all $i \in [n]$ and for all $j \in [\alpha]$ add the vertex $u_{i,j}$ to the graph and connect it to $v_{i,j}$. Let

$$V = \bigcup_{i=1}^{n} V_i$$

and

$$U = \{u_{i,j} \mid i \in [n], j \in [\alpha]\}.$$

Add the vertex set

$$W = \{w_1, \ldots, w_\alpha\}$$

to the graph and for all $j \in [\alpha]$ connect w_j to $v_{1,j}, \ldots, v_{n,j}$.

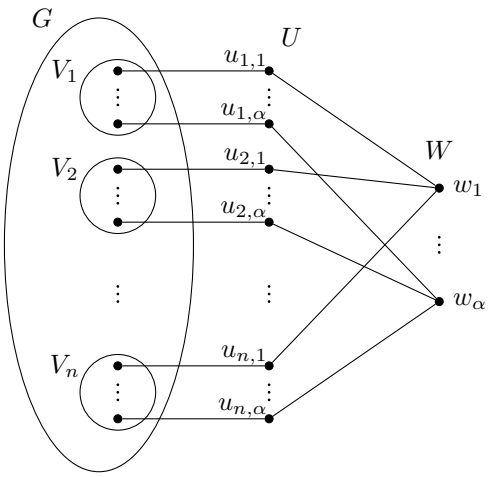


Figure 1: The graph G_α.

We need to prove that G is α-critical with $\alpha(G) = \alpha$ if and only if G_α is minimally 1-tough. First we prove the following lemma.

Lemma 14 *Let G be a graph with $\alpha(G) \leq \alpha$. Then G_α is 1-tough.*

Proof: Let $S \subseteq V(G_\alpha)$ be a cutset. We show that $\omega(G_\alpha - S) \leq |S|$.

Case 1: $W \subseteq S$. If a vertex of U has only one neighbor in $V(G_\alpha) \setminus S$, then we can assume that this vertex is not in S. Then there are two types of components in $G_\alpha - S$: isolated vertices from U and components containing at least one vertex from V. There are at most $\alpha(G)$ components of the second type and (exactly) $|V \cap S| = |S| - \alpha$ components of the first type. Thus $\omega(G_\alpha - S) \leq |S| - \alpha + \alpha(G) \leq |S|$.

Case 2: $W \not\subseteq S$. First, we make two convenient assumptions for S.

(1) $U \cap S = \emptyset$.

It is easy to see that if $u_{i,j} \in S$, then we can assume that $v_{i,j} \notin S$. Now there are two cases.

Case 2.1: $v_{i,j}$ is not isolated in $G_\alpha - S$. Then we can consider $S' = (S \setminus \{u_{i,j}\}) \cup \{v_{i,j}\}$ instead of S.

Case 2.2: $v_{i,j}$ is isolated in $G_\alpha - S$. Since there are no isolated vertices in G, there exists $k \in [n]$ such that $v_i v_k \in E(G)$. Then $v_{k,j} \in S$, so $u_{k,j} \notin S$, which means that w_j is not isolated in $G_\alpha - S$, so we can consider $S' = (S \setminus \{u_{i,j}\}) \cup \{w_j\}$ instead of S.

(2) For all $i \in [n]$, either $V_i \subseteq S$ or $V_i \cap S = \emptyset$.

After the assumption (1), assume that only a proper subset of V_i is contained in S. Let v be an element of this subset. We can consider the cutset $S \setminus \{v\}$ instead of S, since this decreases the number of components by at most one. So we can repeat this procedure until $V_i \cap S = \emptyset$.

So in $G_\alpha - S$ there are isolated vertices from U and one more component containing the remaining vertices of W and V. So there are less than $|V \cap S|$ isolated vertices, thus

$$\omega(G_\alpha - S) \leq |V \cap S| \leq |S|.$$

So G_α is 1-tough. □

We show that G is α-critical with $\alpha(G) = \alpha$ if and only if G_α is minimally 1-tough.

Let us assume that G is α-critical with $\alpha(G) = \alpha$. So by Lemma 14 G_α is 1-tough. Let $e \in E(G_\alpha)$ be an arbitrary edge. If e has an endpoint in U, then this endpoint has degree 2, so $\tau(G_\alpha - e) < 1$. If e does not have an endpoint in U, then it connects two vertices of V. By Lemma 12 $G_\alpha[V]$ is α-critical, so in $G_\alpha[V] - e$ there exists an independent vertex set I of size $\alpha(G) + 1$. Let $S = (V \setminus I) \cup W$. Then $|S| = \big(|V| - \alpha(G) - 1\big) + \alpha = |V| - 1$ and $\omega\big((G_\alpha - e) - S\big) = |V|$, so $\tau(G_\alpha - e) < 1$.

Let us assume that G is not α-critical with $\alpha(G) = \alpha$.

Case 1: $\alpha(G) > \alpha$. Let I be an independent vertex set of size $\alpha(G)$ in $G_\alpha[V]$ and let $S = (V \setminus I) \cup W$. Then $|S| = \big(|V| - \alpha(G)\big) + \alpha < |V|$ and $\omega(G_\alpha - S) = |V|$, so $\tau(G_\alpha) < 1$, which means that G_α is not minimally 1-tough.

Case 2: $\alpha(G) \leq \alpha$. Since G is not α-critical there exists an edge $e \in E(G)$ such that $\alpha(G - e) \leq \alpha$. By Lemma 14 $(G - e)_\alpha$ is 1-tough, but we can obtain $(G - e)_\alpha$ from G_α by edge-deletion, which means that G_α is not minimally 1-tough. □

4 Further results

Theorem 15 *For every positive integer t,* Min-t-Tough *is DP-complete.*

To prove this more general theorem, first we generalize the construction on Figure 1. We follow a similar argument to show that this construction has the required properties. However, due to the more complicated construction, the proof is harder.

The case when $t \leq 1/2$ is also covered in the paper.

Theorem 16 *For every positive rational number $t = a/b \leq 1/2$,* Min-t-Tough *is DP-complete.*

It is shown that Min-1-Tough can be reduced to this problem. The construction and the proof uses different ideas than the previous proofs.

We were not able to prove the DP-completeness for the remaining t values, but we make the following conjecture.

Conjecture 17 Min-t-Tough *is DP-complete for any positive rational number t.*

References

[1] D. Bauer, S. L. Hakimi, and E. Schmeichel, *Recognizing tough graphs is NP-hard*, Discrete Applied Mathematics **28** (1990), 191–195.

[2] V. Chvátal, *Tough graphs and hamiltonian circuits*, Discrete Mathematics **5** (1973), 215–228.

[3] L. Lovász, *Combinatorial problems and exercises*, AMS Chelsea Publishing, Providence, Rhode Island, 2007.

[4] C. H. Papadimitriou and M. Yannakakis, *The Complexity of Facets (and Some Facets of Complexity)*, Journal of Computer and System Sciences **28** (1984), 244–259.

[5] C. H. Papadimitriou and D. Wolfe, *The Complexity of Facets Resolved*, Journal of Computer and System Sciences **37** (1988), 2–13.

Min-sum-max matroid partitioning problem

Yasushi Kawase[1]

Tokyo Institute of Technology, Tokyo, Japan
kawase.y.ab@m.titech.ac.jp

Kei Kimura[2]

Toyohashi University of Technology, Aichi, Japan
kimura@cs.tut.ac.jp

Kazuhisa Makino[3]

Kyoto University, Kyoto, Japan
makino@kurims.kyoto-u.ac.jp

Hanna Sumita[4]

National Institute of Informatics, Tokyo, Japan
JST, ERATO, Kawarabayashi Large Graph Project, Japan
sumita@nii.ac.jp

Abstract: This paper studies a weighted version of the matroid partitioning problem. In the problem, we are given a weighted matroid $(E, \mathcal{I}, w)$ and an integer k, and the goal is to minimize $\sum_{i=1}^{k} \max_{e \in I_i} w(e)$ among partitions $(I_1, \ldots, I_k)$ of E such that $I_i \in \mathcal{I}$ for all i. We first prove that the problem is strongly NP-hard even if the given matroid is graphic. Then we provide a polynomial-time approximation scheme for the problem.

Keywords: Matroids, Partitioning problem, PTAS

1 Introduction

The *matroid partitioning problem* is one of the most fundamental problems in combinatorial optimization. In this problem, we are given a matroid $(E, \mathcal{I})$ and our task is to find a partition $(I_1, \ldots, I_k)$ of E such that $I_i \in \mathcal{I}$ for all i. We say that such a partition $(I_1, \ldots, I_k)$ of E is *feasible*. The matroid partitioning problem has been eagerly studied in a flow of investigating structures of matroids. See, e.g., [5, 6, 7, 11, 15] for details. In this paper, we study a weighted version of the matroid partitioning problem, which we call the *minimum* $(\sum, \max)$*-value matroid partitioning problem*.

In the problem, we are given a weighted matroid $(E, \mathcal{I}, w)$ and an integer k. For any partition $P = (I_1, \ldots, I_k)$ of E, we call $\sum_{i=1,\ldots,k} \max_{e \in I_i} w(e)$ the $(\sum, \max)$*-value* of P. The minimum $(\sum, \max)$-value matroid partitioning problem is the problem of finding a feasible partition with minimum $(\sum, \max)$-value.

We give an application of the problem that arises in scheduling on identical machines. Suppose that we are given a set E of n jobs and k identical machines, and that each job needs to be allocated on exactly one machine. In addition, we are given size $s(e)$ of job $e \in E$. The set of feasible allocation for each machine is represented by a family $\mathcal{I}$ of independent sets of a matroid. For example, if we can allocate at most 3 jobs on each machine, then $\mathcal{I} = \{X \subseteq E : |X| \leq 3\}$. When a job set I_i is allocated to machine i, then the machine needs $\max_{e \in I_i} s(e)$ units of memory to process all jobs in I_i. The goal of the problem is to minimize the total memory needed, i.e., $(\sum, \max)$-value $\sum_{i=1,\ldots,k} \max_{e \in I_i} s(e)$.

[1]This research was supported by JSPS KAKENHI Grant Number JP16K16005.
[2]This research was supported by JSPS KAKENHI Grant Number JP15H06286.
[3]This research was supported by JSPS KAKENHI Grant Number JP24106002, JP25280004, JP26280001, and JST CREST Grant Number JPMJCR1402, Japan.
[4]This research was supported by JST ERATO Grant Number JPMJER1201, Japan.

Our results

Our main result is to analyze the computational complexity of the minimum $(\sum, \max)$-value matroid partitioning problem. We first show that the problem is strongly NP-hard. At the same time, we also propose a *polynomial-time approximation scheme* (PTAS), i.e., a polynomial-time algorithm that outputs a $(1+\varepsilon)$-approximate solution for any fixed $\varepsilon > 0$. Our PTAS computes an approximate solution by two steps: guess the maximum weight in each I_i^* for an optimal solution $(I_1^*, \ldots, I_k^*)$, and check whether or not there exists such a feasible partition. We remark that the number of possible combinations of the maximum weights is $|E|^k$ and it may be too large. To reduce the possibilities, we use rounding techniques in the design of the PTAS. First, we guess the maximum weight in I_i^* for only s indices (s is set later). Furthermore, we round the weight of each element to reduce the number of different weights to a small number r, which is set later. Then, we now have r^s possibilities. To guarantee the approximation ratio $(1+\varepsilon)$, we need to set r and s to be $\Omega(\log k)$ respectively, and hence the number of possibilities r^s is too large. Our idea to overcome this is to enumerate *sequences* of the maximum weights in the nonincreasing order. This enables us to reduce the number of possibilities to $\binom{r+s-1}{r}$ $(\leq 2^{r+s-1})$, which leads to that our algorithm is a PTAS.

Related work

Burkard and Yao [2] introduced a subclass of matroids including partition matroids, and showed that this class of the minimum $(\sum, \max)$-value matroid partitioning problem can be solved by a greedy algorithm.

The matroid partitioning problems with other objective functions have been studied. One is the problem of finding a feasible partition $(I_1, \ldots, I_k)$ minimizing $\sum_{i=1,\ldots,k} \sum_{e \in I_i} w_i(e)$ when we are given k weights $w_1, \ldots, w_k$. It is known that this problem can be reduced to the *weighted matroid intersection problem*, and vice versa [6]. Here, the weighted matroid intersection problem is to find a maximum weight subset that is simultaneously independent in two given matroids. Many papers have worked on algorithmic aspects of this problem, and in particular, this problem is polynomially solvable (see [11, 15] and references therein). Another such problem is to find a feasible partition minimizing $\max_{i=1,\ldots,k} \sum_{e \in I_i} w(e)$. This problem have been extensively addressed in the scheduling literature under the name of the minimum makespan scheduling. Since this problem is NP-hard, many papers have proposed polynomial-time approximation algorithms. We remark that most researches focused on subclasses of matroids as inputs: for example, free matroids [12, 16], partition matroids [18, 17, 13], uniform matroid [10, 1, 4], and general matroids [17]. Approximation algorithms for the problem of maximizing $\min_{i=1,\ldots,k} \sum_{e \in I_i} w(e)$ are also well-studied [3, 9, 18, 13].

2 Preliminaries

A *matroid* is a set system $(E, \mathcal{I})$ with the following properties: (I1) $\emptyset \in \mathcal{I}$, (I2) $X \subseteq Y \in \mathcal{I}$ implies $X \in \mathcal{I}$, and (I3) $X, Y \in \mathcal{I}$, $|X| < |Y|$ implies the existence of $e \in Y \setminus X$ such that $X \cup \{e\} \in \mathcal{I}$. An inclusion-wise maximal independent set is called a *base*. We denote the set of bases of $(E, \mathcal{I})$ by $B(\mathcal{I})$. All bases of a matroid have the same cardinality, which is called the *rank* of the matroid and is denoted by $\mathrm{rank}(\mathcal{I})$. For any $B_1, B_2 \in B(\mathcal{I})$ and $e_1 \in B_1 \setminus B_2$, there exists $e_2 \in B_2 \setminus B_1$ such that $B_1 - e_1 + e_2 \in B(\mathcal{I})$ and $B_2 - e_2 + e_1 \in B(\mathcal{I})$.

For a matroid $(E, \mathcal{I})$, define

$$\mathcal{I}|A = \{X : A \supseteq X \in \mathcal{I}\}, \qquad \mathcal{I} \setminus A = \{X \setminus A : X \in \mathcal{I}\},$$
$$\mathcal{I}/A = \{X \subseteq E \setminus A : \mathrm{rank}(X \cup A) - \mathrm{rank}(A) = |X|\}, \qquad \mathcal{I}^{(l)} = \{X \in \mathcal{I} : |X| \leq l\}$$

for a subset $A \subseteq E$ and a nonnegative integer $l \in \mathbb{Z}_+$. We call $(A, \mathcal{I}|A)$, $(E \setminus A, \mathcal{I} \setminus A)$, $(E \setminus A, \mathcal{I}/A)$, and $(E, \mathcal{I}^{(l)})$, respectively, the *restriction*, *deletion*, *contraction*, and *truncation* of $(E, \mathcal{I})$. It is well known that $(A, \mathcal{I}|A)$, $(A, \mathcal{I} \setminus A)$, $(E \setminus A, \mathcal{I}/A)$, and $(E, \mathcal{I}^{(l)})$ are all matroids. Given matroids $\mathcal{M}_1 = (E_1, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E_2, \mathcal{I}_2)$, we define the *matroid union*, denoted by $\mathcal{M}_1 \vee \mathcal{M}_2$, to be $(E_1 \cup E_2, \mathcal{I}_1 \vee \mathcal{I}_2)$ where

$\mathcal{I}_1 \vee \mathcal{I}_2 = \{I_1 \cup I_2 : I_1 \in \mathcal{I}_1,\ I_2 \in \mathcal{I}_2\}$. Any matroid union is also a matroid. For more details of matroids, see, e.g., [14].

2.1 Model

Throughout the paper, we assume that every matroid is given by an independence oracle. Let k be a positive integer. We denote $[k] = \{1, \ldots, k\}$. Let $(E, \mathcal{I})$ be a matroid and let $w : E \to \mathbb{R}_+$ be a nonnegative weight. We denote $n = |E|$. For any k sets $I_1, \ldots, I_k \subseteq E$, we call $(I_1, \ldots, I_k)$ a *feasible partition* of E if it satisfies that $\bigcup_{i\in[k]} I_i = E$, $I_i \neq \emptyset$ $(\forall i \in [k])$, $I_i \cap I_j = \emptyset$ $(\forall i, j \in [k],\ i \neq j)$, and $I_i \in \mathcal{I}$ $(\forall i \in [k])$. We define the $(\sum, \max)$-value of a feasible partition $(I_1, \ldots, I_k)$ as

$$\sum_{i\in[k]} \max_{e\in I_i} w(e).$$

In this article, we study the following minimization problem:

$$\min_{(I_1,\ldots,I_k):\ \text{feasible partition}} \sum_{i\in[k]} \max_{e\in I_i} w(e).$$

We refer to the problem as the *minimum* $(\sum, \max)$*-value matroid partitioning problem*. We write a problem instance as $(E, \mathcal{I}, w; k)$.

It is known that we can easily decide whether there exists a feasible partition $(I_1, \ldots, I_k)$ or not via the matroid intersection problem. We observe that checking the feasibility is still easy even for a general setting where each I_i $(i = 1, \ldots, k)$ must belong to a different family $\mathcal{I}_i$ of matroid independent sets. This fact is useful to show our results later.

Theorem 1 (Edmonds [6]) *Let $(E, \mathcal{I}_i)$ be a matroid for $i = 1 \ldots, k$. There exists a polynomial-time algorithm that finds a partition $(I_1, \ldots, I_k)$ of E such that $I_i \in \mathcal{I}_i$ for all i.*

A partition $(I_1, \ldots, I_k)$ is said to be a *base partition* if it is a feasible partition and $I_i \in B(\mathcal{I})$ for all $i \in [k]$. We observe that we only need to consider base partitions. Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid. We add dummy elements so that any feasible partition is a base partition. To describe this precisely, we denote $r = k\cdot\mathrm{rank}(\mathcal{I}) - |E|$. We remark that $r \geq 0$ if E has a feasible partition, since $|E| = \sum_{i\in[k]} |I_i| \leq k\cdot\mathrm{rank}(\mathcal{I})$ holds for any feasible partition $(I_1, \ldots, I_k)$. Then let $D = \{d_1, \ldots, d_r\}$ be a set of dummy elements. Note that $E \cap D = \emptyset$. We define two matroids $\mathcal{M}' = (D, \mathcal{I}')$ and $\overline{\mathcal{M}} = (E \cup D, \overline{\mathcal{I}})$ for each $i \in [k]$ by $\mathcal{I}' = \{D' \subseteq D : |D'| \leq \mathrm{rank}(\mathcal{I}) - 1\}$ and $\overline{\mathcal{I}} = \{I \cup D' : I \in \mathcal{I},\ D' \in \mathcal{I}',\ |I \cup D'| \leq \mathrm{rank}(\mathcal{I})\}$. Namely, $\mathcal{M}'$ is a uniform matroid of rank $(\mathrm{rank}(\mathcal{I}) - 1)$, and $\overline{\mathcal{M}}$ is the $\mathrm{rank}(\mathcal{I})$-truncation of the matroid union $\mathcal{M} \vee \mathcal{M}'$. Then, we have the following proposition.

Proposition 2 *For any $(E, \mathcal{I}, w; k)$, its minimum $(\sum, \max)$-value is the same as the minimum $(\sum, \max)$-value for $(E \cup D, \overline{\mathcal{I}}, \overline{w}; k)$, where*

$$\overline{w}(e) = \begin{cases} w(e) & (e \in E), \\ \min_{e\in E} w(e) & (e \in D). \end{cases}$$

Proof: We observe that by the definition of $\overline{w}$, we have $\max_{e\in I_i} w(e) = \max_{e\in I_i\cup D_i} \overline{w}(e)$ for any $i \in [k]$, $I_i \subseteq E$ and $D_i \subseteq D$ such that $|I_i| \geq 1$. Suppose that $(I_1, \ldots, I_k)$ attains the minimum $(\sum, \max)$-value for $(E, \mathcal{I}, w; k)$ and $(\overline{I}_1, \ldots, \overline{I}_k)$ attains the minimum $(\sum, \max)$-value for $(E \cup D, \overline{\mathcal{I}}, \overline{w}; k)$. Let $I'_i = I_i \cup \{d_l : \sum_{j=1}^{i-1}(\mathrm{rank}(\mathcal{I}) - |I_j|) < l \leq \sum_{j=1}^{i}(\mathrm{rank}(\mathcal{I}) - |I_j|)\}$ and $\overline{I}'_i = \overline{I}_i \setminus D$. Then, since $(\overline{I}'_1, \ldots, \overline{I}'_k)$ is a feasible partition of E with respect to $(E, \mathcal{I}, w; k)$, we have

$$\sum_{i\in[k]} \max_{e\in\overline{I}_i} \overline{w}(e) = \sum_{i\in[k]} \max_{e\in\overline{I}'_i} w(e) \geq \sum_{i\in[k]} \max_{e\in I_i} w(e).$$

On the other hand, since $(I'_1, \ldots, I'_k)$ is a feasible partition of $E \cup D$ with respect to $(E \cup D, \overline{\mathcal{I}}, \overline{w}; k)$, we have

$$\sum_{i\in[k]} \max_{e\in I_i} w(e) = \sum_{i\in[k]} \max_{e\in I'_i} \overline{w}(e) \geq \sum_{i\in[k]} \max_{e\in \overline{I}_i} \overline{w}(e).$$

Thus, we obtain $\sum_{i\in[k]} \max_{e\in I_i} w(e) = \sum_{i\in[k]} \max_{e\in \overline{I}_i} \overline{w}(e)$ and the proposition holds. □

3 Strongly NP-hardness

We prove that the minimum $(\sum, \max)$-value matroid partitioning problem is strongly NP-hard.

To prove this, we use the *densest l-subgraph* problem, which is known to be NP-hard [8]. The densest l-subgraph problem is, given a graph G and an integer l, to find a subgraph of G induced on l vertices that contains the largest number of edges.

In our reduction, we use the following property on a partition matroid. Let $(E, \mathcal{I})$ be a partition matroid defined by $\mathcal{I} = \{I : |I \cap S_i| \leq \eta_i \ (i \in [p])\}$, where $(S_1, \ldots, S_p)$ is a partition of E, and $\eta_1, \ldots, \eta_p$ are positive integers. In addition, we assume that $|S_i| = \eta_i \cdot k$ for each $i \in [p]$ so that E can be partitioned into k bases of $\mathcal{I}$. Then, for any weight w, we can construct greedily an optimal partition to the instance $(E, \mathcal{I}, w; k)$ of the minimum $(\sum, \max)$-value matroid partitioning problem.

Lemma 3 (Burkard and Yao [2]) *Let $(E, \mathcal{I})$ be any partition matroid with $|S_i| = \eta_i \cdot k$ $(\forall i \in [p])$, and let w be any weight. Let $I_{i,j}$ consist of η_i elements with the η_i largest weights in $S_i \setminus (\bigcup_{h=1}^{j-1} I_{i,h})$. Then $(\bigcup_{i\in[p]} I_{i,1}, \ldots, \bigcup_{i\in[p]} I_{i,k})$ is an optimal solution to $(E, \mathcal{I}, w; k)$.*

Proof: Let $(I^*_1, \ldots, I^*_k)$ be an optimal partition. Without loss of generality, we may assume that $\max_{e\in I^*_1} w(e) \geq \cdots \geq \max_{e\in I^*_k} w(e)$. Let j be any index in $[k]$. In addition, let (i', e_j) be the pair of an index and an element attaining $\max_{i\in[p]} \max_{e\in I_{i,j}} w(e)$. We claim that $\max_{e\in I^*_j} w(e) \geq w(e_j)$. To show this, we suppose the contrary. We denote $S = \bigcup_{h<j} I_{i',h} \cup \{e_j\}$. Note that $|S| = (j-1)\eta_{i'} + 1$ and $w(e) \geq w(e_j)$ for all $e \in S$. Since $(E, \mathcal{I})$ is a partition matroid, at most $(j-1)\eta_{i'}$ elements in S are contained in $I^*_1, \ldots, I^*_{j-1}$. By assumption $\max_{e\in I^*_j} w(e) < w(e_j)$, there is an index $\ell > j$ such that I^*_ℓ has some element $e' \in I^*_\ell \cap S$. Then we have $\max_{e\in I^*_\ell} w(e) \geq w(e') \geq w(e_j) > \max_{e\in I^*_j} w(e)$, which contradicts the assumption $\max_{e\in I^*_j} w(e) \geq \max_{e\in I^*_\ell} w(e)$.

Thus, we have

$$\max_{e\in I^*_j} w(e) \geq w(e_j) = \max_{i\in[p]} \max_{e\in I_{i,j}} w(e) = \max_{e\in\bigcup_{i\in[p]} I_{i,j}} w(e).$$

Therefore, $(\bigcup_{i\in[p]} I_{i,1}, \ldots, \bigcup_{i\in[p]} I_{i,k})$ is also an optimal solution. □

Theorem 4 *The minimum $(\sum, \max)$-value matroid partitioning problem is strongly NP-hard.*

Proof: Let $G = (V, F)$ be an instance of the densest l-subgraph problem. We denote $V = \{1, \ldots, n\}$, $F = \{f_1, \ldots, f_m\}$, and $f_i = \{u_i, v_i\}$. For any vertex set $T \subseteq V$, we denote $F[T] = \{\{u, v\} \in F : \{u, v\} \subseteq T\}$.

To solve the densest l-subgraph problem, it suffices to find a set of $n - l$ vertices such that the set of the other l vertices attain $\max_{T\subseteq V} |F[T]|$. We construct a matroid so that every feasible partition of the ground set corresponds to some set of $n - l$ vertices in V, and the $(\sum, \max)$-value is the number of edges in the induced subgraph by the other l vertices.

Let $V' = \{n+1, \ldots, n+2m\}$ be a set of dummy vertices. For each $i \in V \cup V'$, we define a set E_i of $n + 2m - 1$ elements as

$$E_i = \{e_{ij} : j \in \{1, \ldots, n + 2m - 1\}\}.$$

Let

$$E = \bigcup_{i=1}^{n+2m} E_i \quad \text{and} \quad \mathcal{I} = \{I \subseteq E : |I| \leq n + 2m - 1,\ |I \cap E_i| \leq 1\ (\forall i \in [n + 2m])\}.$$

The resulting matroid is denoted by $(E, \mathcal{I})$, which is a $(n + 2m - 1)$-truncation of a partition matroid. We remark that it is also a graphic matroid. We set $k = n + 2m$. The weights of elements are defined as follows:

- for each $j = 1, \ldots, l - 1$, set $w(e_{ij}) = 0 \quad (\forall i \in [n + 2m])$;
- for each $j = l + 2m, \ldots, n + 2m - 1$, set $w(e_{ij}) = m$ if $i \leq n$, and $w(e_{ij}) = 2m^2$ if $i \geq n + 1$;
- set $w(e_{ij})$ $(j = l, l + 1, \ldots, l + 2m - 1)$ as follows: for each $f_t = \{u_t, v_t\}$ $(t = 1, \ldots, m)$,

$$w(e_{i,l+2t-2}) = \begin{cases} t - 1 & (i \in \{1, \ldots, n\}), \\ 0 & (i \in \{n + 1, \ldots, n + 2m\}), \end{cases}$$

and

$$w(e_{i,l+2t-1}) = \begin{cases} t & (i \in \{u_t, v_t\}), \\ t - 1 & (i \in \{1, \ldots, n\} \setminus \{u_t, v_t\}), \\ 0 & (i \in \{n + 1, \ldots, n + 2m\}). \end{cases}$$

The weight is illustrated in Table 1.

Table 1: The weight of each element e_{ij}, where each column corresponds to i and each row corresponds to j.

$j \backslash i$	1	$\cdots$	$\cdots$	u_t	$\cdots$	$\cdots$	$\cdots$	v_t	$\cdots$	$\cdots$	n	$n+1$	$\cdots$	$n+2m$
1	0	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	0	0	$\cdots$	0
	$\vdots$										$\vdots$	$\vdots$		$\vdots$
$l-1$	0	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	0	0	$\cdots$	0
l	0	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	0	0	$\cdots$	0
												$\vdots$		$\vdots$
$l+2t-2$	$t-1$	$\cdots$	$t-1$	$t-1$	$t-1$	$\cdots$	$t-1$	$t-1$	$t-1$	$\cdots$	$t-1$	0	$\cdots$	0
$l+2t-1$	$t-1$	$\cdots$	$t-1$	t	$t-1$	$\cdots$	$t-1$	t	$t-1$	$\cdots$	$t-1$	0	$\cdots$	0
$l+2t$	t	$\cdots$	t	t	t	$\cdots$	t	t	t	$\cdots$	t	0	$\cdots$	0
												$\vdots$		$\vdots$
$l+2m-1$												0	$\cdots$	0
$l+2m$	m	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	m	$2m^2$	$\cdots$	$2m^2$
	$\vdots$										$\vdots$	$\vdots$		$\vdots$
$n+2m-1$	m	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	m	$2m^2$	$\cdots$	$2m^2$

We remark that $|E| = (n + 2m)(n + 2m - 1)$. By the definition of the matroid, for every $i \in [n + 2m]$, all elements in E_i belong to different independent sets from each other. Thus, for any feasible partition of E, each independent set has $n + 2m - 1$ elements which consist of one element from each E_i except one set.

It remains to show that the resulting instance is equivalent to the densest l-subgraph problem instance $(G = (V, F), l)$.

Claim 5 *Let $\alpha \in \{0, \ldots, m\}$. The graph G has a vertex set T^* with $|T^*| = l$ and $|F[T^*]| \geq \alpha$ if and only if there exists a feasible partition $(I_1, \ldots, I_k)$ of E with $(\sum, \max)$-value at most $2m^2(n - l) + m^2 + m - \alpha$.*

First, we assume that there exists $T^* \subseteq V$ such that $|T^*| = l$ and $|F[T^*]| \geq \alpha$. Without loss of generality, we assume that $T^* = \{1, \ldots, l\}$ and $V \setminus T^* = \{l+1, \ldots, n\}$. We show that there exists a partition such that its $(\sum, \max)$-value is at most $2m^2(n-l) + m^2 + m - \alpha$. We denote $E^j[p,q] = \{e_{p,j}, \ldots, e_{q,j}\}$. Let $J_1 = \{1, \ldots, l\}$, $J_2 = \{l+1, \ldots, l+2m\}$, and $J_3 = \{l+2m+1, \ldots, n+2m\}$. We construct a partition $(I_1^*, \ldots, I_{n+2m}^*)$ of E as follows:

$$I_j^* = \begin{cases} E^{j-1}[1, j-1] \cup E^j[j+1, n+2m] & (j \in J_1), \\ E^{j-1}[1, l] \cup E^j[l+1, n+2m+l-j] \cup E^{j-1}[n+2m+l-j+2, n+2m] & (j \in J_2), \\ E^{j-1}[1, j-2m-1] \cup E^j[j-2m+1, n] \cup E^{j-1}[n+1, n+2m] & (j \in J_3). \end{cases}$$

Then, the maximum weight of each independent set is

$$\max_{e \in I_j^*} w(e) = \begin{cases} 0 & (j \in J_1), \\ t-1 & (j = l+2t-1 \in J_2,\ t = 1, \ldots, m,\ \{u_t, v_t\} \in F[T^*]), \\ t & (j = l+2t-1 \in J_2,\ t = 1, \ldots, m,\ \{u_t, v_t\} \notin F[T^*]), \\ t & (j = l+2t \in J_2,\ t = 1, \ldots, m), \\ 2m^2 & (j \in J_3). \end{cases}$$

Thus, the $(\sum, \max)$-value is at most

$$0 \cdot l + \sum_{t=1}^{m} (2t) - |F[T^*]| + 2m^2 \cdot (n-l) \leq 2m^2(n-l) + m^2 + m - \alpha.$$

Conversely, we assume that there exists a feasible partition $(I_1, \ldots, I_k)$ of E such that $\max_{e \in I_1} w(e) \leq \cdots \leq \max_{e \in I_k} w(e)$, and

$$\sum_{j \in [k]} \max_{e \in I_j} w(e) \leq 2m^2(n-l) + m^2 + m - \alpha.$$

All elements in E_k must be contained in different I_j's from each other by definition of $(E, \mathcal{I})$. Hence at least $n-l$ sets contain elements e with $w(e) = 2m^2$. If $\max_{e \in I_j} w(e) \geq 2m^2$ holds for some $j \leq l+2m$, then the objective value is at least $2m^2(n-l+1) > 2m^2(n-l) + m^2 + m - \alpha$. Thus, each of $I_{l+2m+1}, \ldots, I_k$ contains $2m$ elements with weight $2m^2$, and none of $I_1, \ldots, I_{l+2m}$ contains such elements. Let

$$U = \{i : |E_i \cap I_j| = 0\ (\exists j \in \{l+2m+1, \ldots, k\})\}.$$

Note that $|U| = n-l$ and $U \subseteq \{1, \ldots, n\}$. Here, we have

$$\begin{aligned} 2m^2(n-l) + m^2 + m - \alpha &\geq \sum_{j \in [k]} \max_{e \in I_j} w(e) \\ &= 2m^2(n-l) + \sum_{j \in [l+2m]} \max_{e \in I_j} w(e). \end{aligned}$$

In order to obtain a lower bound of $\sum_{j \in [l+2m]} \max_{e \in I_j} w(e)$, we define $E' = \{e_{ij} : i \in U,\ j = 1, \ldots, l+2m\}$. Let $(E', \mathcal{I}')$ be a partition matroid where $\mathcal{I}' = \{I' : |I' \cap E_i| \leq 1\ (\forall i \in U)\}$. We observe that $\sum_{j \in [l+2m]} \max_{e \in I_j} w(e) \geq \sum_{j \in [l+2m]} \max_{e \in I_j \cap E'} w(e)$, and $(I_1 \cap E', \ldots, I_{l+2m} \cap E')$ is a feasible partition to the $(\sum, \max)$ problem instance $(E', \mathcal{I}', w; l+2m)$. By Lemma 3, an optimal solution to $(E', \mathcal{I}', w; l+2m)$ can be obtained by a greedy algorithm. Let $(I_1', \ldots, I_{l+2m}')$ be an output solution of the greedy algorithm. Then we have

$$\begin{aligned} \sum_{j \in [l+2m]} \max_{e \in I_j} w(e) \geq \sum_{j \in [l+2m]} \max_{e \in I_j'} w(e) &= m + \sum_{l=1}^{m} 2(l-1) + |\{\{u,v\} : |\{u,v\} \cap U| \geq 1\}| \\ &\geq m^2 + m - |F[V \setminus U]|. \end{aligned}$$

This implies $|F[V \setminus U]| \geq \alpha$. Therefore, $T = V \setminus U$ is a vertex set with $|T| = l$ and $|F[T]| \geq \alpha$.

This proves the theorem. $\square$

4 Polynomial-time approximation scheme

In this section, we provide a PTAS for the minimum $(\sum, \max)$-value matroid partitioning problem. This is the best possible result (unless P=NP) because the problem is strongly NP-hard as we proved in the previous section.

We start with the following observation.

Proposition 6 *Let $(E, \mathcal{I}, w; k)$ be any instance of the minimum $(\sum, \max)$-value matroid partitioning problem, and let $(I_1^*, \ldots, I_k^*)$ be an optimal solution. When we know $\max_{e \in I_i^*} w(e)$ for all $i \in [k]$, we can easily compute a feasible partition $(I_1, \ldots, I_k)$ such that $\sum_{i \in [k]} \max_{e \in I_i} w(e) \leq \sum_{i \in [k]} \max_{e \in I_i^*} w(e)$.*

Proof: The feasible partitions for matroids $(E, \mathcal{I}|\{e : w(e) \leq \max_{e^* \in I_i^*} w(e^*)\})_{i \in [k]}$ satisfy the condition. Thus, we can find one of them in polynomial-time by Theorem 1. □

Let $(E, \mathcal{I}, w; k)$ be a problem instance, and let $\varepsilon < 1/2$ be a positive number. We write $w^{\max} = \max_{e \in E} w(e)$. Let $(I_1^*, \ldots, I_k^*)$ be an optimal solution.

The idea of the algorithm is to guess the maximum weights. In order to reduce the number of possibilities, we guess $\max_{e \in I_i^*} w(e)$ only for some i's. Without loss of generality, we assume that $\max_{e \in I_1^*} w(e) \geq \cdots \geq \max_{e \in I_k^*} w(e)$. We define a set $J = \{i_1, \ldots, i_s\}$ of indices by

$$i_j = \begin{cases} j & (j = 1, \ldots, \lfloor 1/\varepsilon^2 \rfloor), \\ \lfloor (1+\varepsilon)^t / \varepsilon^2 \rfloor & (j = \lfloor 1/\varepsilon^2 \rfloor + t,\ t = 1, \ldots, \lfloor \log_{1+\varepsilon}(k\varepsilon^2) \rfloor). \end{cases}$$

By definition, it holds that $1 = i_1 < i_2 < \cdots < i_s \leq k$, and $s = \lfloor 1/\varepsilon^2 \rfloor + \lfloor \log_{1+\varepsilon}(k\varepsilon^2) \rfloor$. Note that for any $j = \lfloor 1/\varepsilon^2 \rfloor + t$ and $t \geq 1$, we have

$$i_j - i_{j-1} \geq ((1+\varepsilon)^t/\varepsilon^2 - 1) - ((1+\varepsilon)^{t-1}/\varepsilon^2) = (1+\varepsilon)^{t-1}/\varepsilon - 1 \geq 1/\varepsilon - 1 > 1$$

as $\varepsilon < 1/2$. For notational convenience, we denote $i_0 = 0$ and $i_{s+1} = k + 1$.

To reduce the number of possibilities more, we round the weights $w(e)$. For all $e \in E$, define

$$w'(e) = \begin{cases} \frac{(1+\varepsilon)^t w^{\max}}{k}\varepsilon & \left(\frac{(1+\varepsilon)^t w^{\max}}{k}\varepsilon \leq w(e) < \frac{(1+\varepsilon)^{t+1} w^{\max}}{k}\varepsilon,\ t = 0, 1, \ldots, \lfloor \log_{1+\varepsilon} \frac{k}{\varepsilon} \rfloor\right), \\ 0 & \left(w(e) < \frac{w^{\max}}{k}\varepsilon\right). \end{cases}$$

Our algorithm guesses $\max_{e \in I_{i_j}^*} w'(e)$ for each $i_j \in J$. We write u_j^* for the value. Then, it finds a feasible partition $(I_1, \ldots, I_k)$ that satisfies $\max_{e \in I_1} w(e) \geq \cdots \geq \max_{e \in I_k} w(e)$ and $\max_{e \in I_{i_j}} w'(e) \leq u_j^*$ for all $i_j \in J$. The algorithm is summarized in Algorithm 1.

Algorithm 1: PTAS for the $(\sum, \max)$ problem

1 **foreach** $u_1, \ldots, u_s \in \{0\} \cup \left\{ \frac{(1+\varepsilon)^t w^{\max}}{k}\varepsilon : t = 0, \ldots, \lfloor \log_{1+\varepsilon}(k/\varepsilon) \rfloor \right\}$ *such that* $u_1 \geq \cdots \geq u_s$ **do**

2 find a partition $(I_1, \ldots, I_k)$ such that $I_i \in (\mathcal{I}|\{e : w'(e) \leq u_j\})$ for each $i_j \leq i < i_{j+1},\ j = 1, \ldots, s$ if exists;

3 **return** the best solution $(I_1, \ldots, I_k)$ among the obtained partitions;

In what follows, we prove that Algorithm 1 is a PTAS for the minimum $(\sum, \max)$-value matroid partitioning problem.

Let $(I_1^*, \ldots, I_k^*)$ be an optimal solution to the problem and $(I_1, \ldots, I_k)$ be the output of Algorithm 1. Without loss of generality, we assume that $\max_{e \in I_1^*} w(e) \geq \cdots \geq \max_{e \in I_k^*} w(e)$. Let $u_j^* = \max_{e \in I_{i_j}^*} w'(e)$ for each $i_j \in J$.

We first analyze the running time of Algorithm 1.

Claim 7 *Algorithm 1 runs in polynomial-time with respect to k for fixed ε.*

PROOF: Let $r = \lfloor \log_{1+\varepsilon}(k/\varepsilon) \rfloor + 2$. We observe that any choice of a possible combination of values $u_1, \ldots, u_s$ corresponds a multisubset of size s from the set of r values. Thus the number of possible combinations is $\binom{r+s-1}{s}$. Furthermore, we have

$$\begin{aligned}\binom{r+s-1}{s} &\leq \sum_{l=0}^{r+s-1} \binom{r+s-1}{l} = 2^{r+s-1} \leq 2^{(\log_{1+\varepsilon}(k/\varepsilon)+2)+(1/\varepsilon^2+\log_{1+\varepsilon}(k\varepsilon^2))} \\ &\leq 2^{2\log_{1+\varepsilon} k+2+1/\varepsilon^2} = 2^{2+1/\varepsilon^2} \cdot k^{\log_{1+\varepsilon} 4}.\end{aligned}$$

This is a polynomial with respect to k for fixed ε. Thus, the algorithm runs in polynomial-time. □

Note that, without the restriction $u_1 \geq \cdots \geq u_s$, the number of possible combinations of values $u_1, \ldots, u_s$ is $r^s = k^{\Theta(\log \log k)}$, which is not polynomial with respect to k.

In the remainder, we show the approximation ratio of the algorithm.

Claim 8 *Let* OPT *denote the optimal value* OPT *and let* ALG *denote the* $(\sum, \max)$*-value of* $(I_1, \ldots, I_k)$*. Then it holds that* $\mathrm{ALG} \leq (1 + 15.5\varepsilon)\mathrm{OPT}$*.*

PROOF: First, OPT is at least

$$\mathrm{OPT} = \sum_{i \in [k]} \max_{e \in I_i^*} w(e) \geq \sum_{i \in [k]} \max_{e \in I_i^*} w'(e) \geq \sum_{j=1}^{s} (i_j - i_{j-1}) u_j^*.$$

Let $(I_1', \ldots, I_k')$ be a feasible partition of E obtained at line 2 in Algorithm 1 using $u_1^*, \ldots, u_s^*$. Then ALG is at most

$$\begin{aligned}\mathrm{ALG} &= \sum_{i \in [k]} \max_{e \in I_i} w(e) \leq \sum_{i \in [k]} \max_{e \in I_i'} w(e) \\ &\leq \sum_{j=1}^{s} (i_{j+1} - i_j) \max_{e \in I_{i_j}'} w(e) \\ &\leq \sum_{j=1}^{s} (i_{j+1} - i_j) \left((1+\varepsilon) u_j^* + \frac{w^{\max}}{k} \varepsilon \right) \\ &\leq \sum_{j=1}^{s} (i_{j+1} - i_j)(1+\varepsilon) u_j^* + k \cdot \frac{w^{\max}}{k} \varepsilon \leq (1+\varepsilon) \sum_{j=1}^{s} (i_{j+1} - i_j) u_j^* + \varepsilon \cdot \mathrm{OPT}. \qquad (1)\end{aligned}$$

Here, the third inequality holds by the definition of w' and $\max_{e \in I_{i_j}'} w'(e) \leq u_j^*$.

We derive an upper bound on $\sum_{j=1}^{s} (i_{j+1} - i_j) u_j^*$. To simplify notation, let $q = \lfloor 1/\varepsilon^2 \rfloor$. First, since $i_{j+1} - i_j = i_j - i_{j-1} = 1$ holds for any $j = 1, \ldots, q-1$, we have

$$\sum_{j=1}^{q-1} (i_{j+1} - i_j) u_j^* = \sum_{j=1}^{q-1} (i_j - i_{j-1}) u_j^*. \qquad (2)$$

Second, we evaluate $(i_{q+1} - i_q) u_q^*$. Note that $i_q = q = \lfloor 1/\varepsilon^2 \rfloor$ and $i_{q+1} = \lfloor (1+\varepsilon)/\varepsilon^2 \rfloor$. Thus $i_{q+1} - i_q \leq (1+\varepsilon)/\varepsilon^2 - (1/\varepsilon^2 - 1) = (1+\varepsilon)/\varepsilon$. Moreover,

$$u_q^* = \max_{e \in I_q^*} w'(e) \leq \max_{e \in I_q^*} w(e) \leq \mathrm{OPT}/q,$$

because $\mathrm{OPT} = \sum_{i\in[k]} \max_{e\in I_i^*} w(e) \geq \sum_{i\in[q]} \max_{e\in I_i^*} w(e) \geq q \cdot \max_{e\in I_q^*} w(e)$. We remark that $1/q = 1/\lfloor 1/\varepsilon^2 \rfloor \leq 1/(1/\varepsilon^2 - 1) = \varepsilon^2/(1-\varepsilon^2) < \frac{4}{3}\varepsilon^2 < 2\varepsilon^2$ as $\varepsilon < 1/2$. Therefore, it follows that

$$(i_{q+1} - i_q)u_q^* \leq 2\varepsilon(1+\varepsilon)\mathrm{OPT}. \tag{3}$$

Lastly, let $j \in \{q+1, \ldots, s\}$, and let t (≥ 1) be the integer such that $i_j = \lfloor (1+\varepsilon)^t/\varepsilon^2 \rfloor$ (i.e., $t = j-q$). We observe that $i_j - i_{j-1} \geq (1+\varepsilon)^{t-1}/\varepsilon - 1$. In addition, we have

$$\begin{aligned} i_{j+1} - i_j &\leq \left(\frac{(1+\varepsilon)^{t+1}}{\varepsilon^2}\right) - \left(\frac{(1+\varepsilon)^t}{\varepsilon^2} - 1\right) = \frac{(1+\varepsilon)^t}{\varepsilon} + 1 \\ &\leq \frac{(1+\varepsilon)/\varepsilon + 1}{(1+\varepsilon)^0/\varepsilon - 1}\left(\frac{(1+\varepsilon)^{t-1}}{\varepsilon} - 1\right) \\ &\leq \frac{1+2\varepsilon}{1-\varepsilon}(i_j - i_{j-1}) < (1+6\varepsilon)(i_j - i_{j-1}), \end{aligned}$$

where the second inequality holds since $\frac{(1+\varepsilon)^x/\varepsilon+1}{(1+\varepsilon)^{x-1}/\varepsilon-1}$ is monotone decreasing for $x \geq 1$ and the last inequality holds since $\varepsilon < 1/2$. Therefore, it follows that

$$\sum_{j=q+1}^{s} (i_{j+1} - i_j)u_j^* = \sum_{j=q+1}^{s} (1+6\varepsilon)(i_j - i_{j-1})u_j^*. \tag{4}$$

By combining (1)–(4) together with $\varepsilon < 1/2$, we have

$$\begin{aligned} \mathrm{ALG} &\leq (1+\varepsilon)\left((1+6\varepsilon)\sum_{j=1}^{s}(i_j - i_{j-1})u_j^* + 2\varepsilon(1+\varepsilon)\mathrm{OPT}\right) + \varepsilon \cdot \mathrm{OPT} \\ &\leq (1+\varepsilon)\left((1+6\varepsilon) + 2\varepsilon(1+\varepsilon)\right) \cdot \mathrm{OPT} + \varepsilon \cdot \mathrm{OPT} \\ &= (1 + 10\varepsilon + 10\varepsilon^2 + 2\varepsilon^3)\mathrm{OPT} \\ &< (1 + 10\varepsilon + 5\varepsilon + 0.5\varepsilon)\mathrm{OPT} = (1 + 15.5\varepsilon)\mathrm{OPT}. \end{aligned}$$

This proves Claim 8. □

Claims 7 and 8 imply that Algorithm 1 is a PTAS.

Theorem 9 *Algorithm 1 is a PTAS for the minimum* $(\sum, \max)$*-value matroid partitioning problem.*

References

[1] Luitpold Babel, Hans Kellerer, and Vladimir Kotov. The k-partitioning problem. *Mathematical Methods of Operations Research*, 47:59–82, 1998.

[2] Rainer E. Burkard and Enyu Yao. Constrained partitioning problems. *Discrete Applied Mathematics*, 28:21–34, 1990.

[3] Shi Ping Chen, Yong He, and Guohui Lin. 3-partitioning for maximizing the minimum load. *Journal of Combinatorial Optimization*, 6:67–80, 2002.

[4] Mauro Dell'Amico and Silvano Martello. Bounds for the cardinality constrained $P||C_{\max}$ problem. *Journal of Scheduling*, 4:123–138, 2001.

[5] Jack Edmonds. Minimum partition of a matroid into independent subsets. *JOURNAL OF RESEARCH of the National Bureau of Standards—B. Mathematicsnd Mathematical Physics*, 69B:67–72, 1965.

[6] Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and their Applications (Proceedings of the Calgary International Conference on Combinatorial Structures and Their Applications 1969)*, pages 69–87. Gordon and Breach, New York, 1970.

[7] Jack Edmonds and Delbert R. Fulkerson. Transversals and matroid partition. *Journal of Research of the National Bureau of Standards*, 69B:147–153, 1965.

[8] Uriel Feige, David Peleg, and Guy Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[9] Yong He, Zhiyi Tan, Jing Zhu, and Enyu Yao. k-partitioning problems for maximizing the minimum load. *Computers and Mathematics with Applications*, 46:1671–1681, 2003.

[10] Hans Kellerer and Gerhard Woeginger. A tight bound for 3-partitioning. *Discrete Applied Mathematics*, 45:249–259, 1993.

[11] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2002.

[12] Jan K. Lenstra, David B. Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46:259–271, 1990.

[13] Weidong Li and Jianping Li. Approximation algorithms for k-partitioning problems with partition matroid constraint. *Optimization Letters*, 8:1093–1099, 2014.

[14] James G. Oxley. *Matroid Theory*. Oxford University Press, New York, 1992.

[15] Alexander Schrijver. *Combinatorial Optimization*. Springer, 2003.

[16] José Verschae and Andreas Wiese. On the configuration-LP for scheduling on unrelated machines. *Journal of Scheduling*, 17:371–383, 2014.

[17] Biao Wu and Enyu Yao. Lower bounds and modified LPT algorithm for k-partitioning problems with partition matroid constraint. *Applied Mathematics-A Journal of Chinese Universities*, 23:1–8, 2008.

[18] Biao Wu and Enyue Yao. k-partitioning problems with partition matroid constraint. *Theoretical Computer Science*, 374:41–48, 2007.

Reachability-based matroid-restricted packing of arborescences

Csaba Király

Department of Operations Research
Eötvös Loránd University
Pázmány Péter sétány 1/C
Budapest, Hungary, 1117
cskiraly@cs.elte.hu

Zoltán Szigeti

Univ. Grenoble Alpes
G-SCOP
46, Avenue Félix Viallet
Grenoble, France, 38000
zoltan.szigeti@grenoble-inp.fr

Abstract: The fundamental result of Edmonds [5] started the area of packing arborescences and the great number of recent results shows increasing interest of this subject. Two types of matroid constraints were added to the problem in [2, 3, 9], here we show that both contraints can be added simultaneously. This way we provide a solution to a common generalization of the reachability-based packing of arborescences problem of the first author [14] and the matroid intersection problem of Edmonds [4].

Keywords: connectivity, arborescence, packing, matroid

1 Introduction

This paper considers problems on arborescence packings in rooted digraphs where a **rooted digraph** is a digraph $D = (V + s, A)$ with a designated root vertex s. Throughout this paper a **packing** in a digraph means arc-disjoint subgraphs. Different types of matroid constraints will be added simultaneously to the arborescence packing problem in such a way that the problem obtained contains the matroid intersection problem. The solution provided to this problem in this paper applies ideas from the proof of the matroid intersection theorem of Edmonds [4].

Let $D = (V + s, A)$ be a rooted digraph. For $X \subseteq V$, let $\overline{X} = V + s - X$. For $Z \subseteq \overline{X}$, $\boldsymbol{\partial_Z(X)}$ denotes the set of arcs from Z to X. If $Z = \overline{X}$, then Z is omitted from the index. By consequence, $|\partial(X)|$ is the in-degree of the set X.

An ***s*-arborescence** is a directed tree on a vertex-set containing the **root** vertex s in which each vertex has in-degree 1 except s. An s-arborescence in a rooted digraph $D = (V + s, A)$ is **spanning** if its vertex set is $V + s$. For definitions from matroid theory, we refer to the next section.

Edmonds [5] solved the packing problem of k spanning s-arborescences in a rooted digraph. It is well-known that this problem can be formulated as a matroid intersection problem. Indeed, if the first matroid is the k-sum of the graphic matroid of the underlying undirected graph of D and the second matroid is the direct sum of the uniform matroids $U_{|\partial(v)|,k}$ on the set $\partial(v)$ of arcs entering v for $v \in V$, then the set of the arc sets of the union of k arc-disjoint spanning s-arborescences of D is the set of common bases of these two matroids.

Frank [9] (and later Bernáth and T. Király [2]) observed that one can go further, namely in the above construction the uniform matroids can be replaced by arbitrary matroids. It is mentioned in [9] that this way one may get a solution to the problem of matroid-restricted packing of k spanning s-arborescences where a packing of s-arborescences $T_1, \ldots, T_k$ in a rooted digraph $D = (V + s, A)$ is said to be **matroid-restricted** if, given a matroid $\mathcal{M}_v$ on $\partial(v)$ for every $v \in V$,

$$\{A(T_i) \cap \partial(v) : T_i \text{ contains } v\} \text{ is independent in } \mathcal{M}_v \text{ for every } v \in V. \tag{1}$$

If $\mathcal{M}$ is the direct sum of the matroids $\mathcal{M}_v = (\partial(v), r_v)$ for $v \in V$, then a matroid-restricted packing is called an $\mathcal{M}$-restricted packing.

Another way of adding a matroid constraint to the problem of packing arborescences, was proposed by Durand de Gevigney, Nguyen, Szigeti in [3]. A packing of s-arborescences $T_1, \ldots, T_t$ in a rooted digraph $D = (V + s, A)$ is said to be **matroid-based** if, given a matroid $\mathcal{M}$ on the set $\partial(V)$ of arcs leaving s,

$$\{\partial(V) \cap A(T_i[s, v]) : T_i \text{ contains } v\} \text{ is a base of } \mathcal{M} \text{ for every } v \in V, \tag{2}$$

where $\boldsymbol{T[s, v]}$ denotes the unique path from s to v in an s-arborescence T. Durand de Gevigney, Nguyen, Szigeti [3] solved the problem of matroid-based packing of s-arborescences. Bérczi and Frank proposed later a more natural problem of matroid-based packing of *spanning* s-arborescences (see in [1]). Recently, a superset of the authors of this paper in [8] proved that this problem is NP-complete.

We propose in this paper to solve the problem of matroid-based matroid-restricted packing of s-arborescences where both of the above matroid constraints are added. Note that the proposed problem contains the problems of matroid-based packing of s-arborescences and matroid-restricted packing of spanning s-arborescences. It is not surprising that it also contains the problem of matroid intersection. Indeed, if $\mathcal{M}_1$ and $\mathcal{M}_2$ are two matroids on S, then the problem of matroid-based matroid-restricted packing of s-arborescences for the instance of digraph, with two vertices s and v and parallel arcs sv each corresponding to an element of S, and matroids $\mathcal{M}_1$ and $\mathcal{M}_2$, reduces to the matroid intersection problem.

Observe that, by the above mentioned negative result of [8], the problem of matroid-based matroid-restricted packing of spanning s-arborescences is NP-complete, however, we will solve this problem for special cases where the first matroid is restricted to several fundamental classes. Observe that, by [2, Corollary 3.2] and by the matroid intersection algorithm of Edmonds, the problem of matroid-based matroid-restricted packing of spanning s-one-arborescences can be solved in polynomial time where an **s-one-arborescence** is an s-arborescence with only one arc leaving its root s.

In fact, we will propose an even more general problem. To be able to do this, we mention another direction in which the problem of packing spanning arborescences was generalized. Kamiyama, Katoh, Takizawa [13] solved the packing problem of k reachability s-one-arborescences where an s-one-arborescence with a root arc e in a rooted digraph $D = (V + s, A)$ is said to be a **reachability** s-one arborescence if its vertex set is the set of vertices reachable from s by a directed path of D with first arc e.

The first author [14] provided a common generalization of the problems of matroid-based packing of s-arborescences and packing of k reachability s-one-arborescences, namely the problem of reachability-based packing of s-arborescences, where a packing of s-arborescences $T_1, \ldots, T_t$ in a rooted digraph $D = (V + s, A)$ is said to be **reachability-based** if, given a matroid $\mathcal{M}$ on $\partial(V)$ with rank function r,

$$\{\partial(V) \cap A(T_i[s, v]) : T_i \text{ contains } v\} \text{ is independent in } \mathcal{M} \text{ of size } r(\partial_s(P(v))) \text{ for all } v \in V, \tag{3}$$

where $\boldsymbol{P(X)}$ denotes the set of vertices in V from which X is reachable by a directed path in D. Note that, by definition, $P(X)$ contains X and does not contain the vertex s.

In this paper, we will solve the problem of reachability-based matroid-restricted packing of s-arborescences. We will show that, by applying the proof method of the matroid intersection theorem, the problem of reachability-based matroid-restricted packing of s-arborescences can be reduced to the problem of reachability-based packing of s-arborescences.

2 Definitions

We need some basic terminologies from matroid theory, we refer to [10] for more details. A function $b : 2^\Omega \to \mathbb{Z}$ is called **submodular** if for all $X, Y \subseteq \Omega$,

$$b(X) + b(Y) \quad \geq \quad b(X \cap Y) + b(X \cup Y).$$

A function $p : 2^\Omega \to \mathbb{Z}$ is called **supermodular** if $-p$ is submodular. By the results of Iwata, Fleischer and Fujishige [12] and independently by Schrijver [15], a submodular function can be minimized in polynomial time.

For a set function $r : 2^{\mathsf{S}} \to \mathbb{Z}_+$, $\mathcal{M} = (\mathsf{S}, r)$ is called a **matroid** if r is 0 on the $\emptyset$, monotone non-decreasing, subcardinal ($r(\mathsf{Q}) \leq |\mathsf{Q}|$) and submodular. The members of $\mathcal{I} = \{\mathsf{Q} \subseteq \mathsf{S} : r(\mathsf{Q}) = |\mathsf{Q}|\}$ are called **independent** sets of the matroid and r is called the **rank function** of the matroid. It is well known that a matroid can also be defined by its independent sets. Let $\mathsf{Q} \subseteq \mathsf{S}$. The maximal independent sets in Q are called **bases** of Q. Note that all bases of Q are of the same size, namely $r(\mathsf{Q})$. The bases of S are called the bases of $\mathcal{M}$. We say that an element s of Q is a **bridge** of Q if $r(\mathsf{Q} - \mathsf{s}) = r(\mathsf{Q}) - 1$. We define $\mathrm{Span}_{\mathcal{M}}(\mathsf{Q}) = \{\mathsf{s} \in \mathsf{S} : r(\mathsf{Q} \cup \{\mathsf{s}\}) = r(\mathsf{Q})\}$. Note that $\mathrm{Span}_{\mathcal{M}}$ is monotone.

As examples, let us mention the following matroids:

1. **graphic matroid**: $\mathcal{I}$ = edge sets of forests in a graph;
2. **transversal matroid**: $\mathcal{I}$ = subsets of S that can be covered by a matching in a bipartite graph $G = (S, T; E)$;
3. **uniform matroids $U_{n,k}$**: $\mathcal{I} = \{\mathsf{Q} \subseteq \mathsf{S} : |\mathsf{Q}| \leq \mathsf{k}\}$ where $|\mathsf{S}| = n$;
4. **free matroid**: $U_{n,n}$.

Note that uniform matroids form a special class of transversal matroids where G is the complete bipartite graph $K_{n,k}$.

We will need the following operations on matroids. Let $\mathcal{M} = (\mathsf{S}, r)$ be a matroid. For $\mathsf{Q} \subseteq \mathsf{S}$, $\boldsymbol{\mathcal{M}|_{\mathsf{Q}}}$ is the matroid with rank function $\boldsymbol{r|_{\mathsf{Q}}}$ obtained from $\mathcal{M}$ by restriction on Q. For $\mathsf{s} \in \mathsf{S}$, $\boldsymbol{\mathcal{M} - \mathsf{s}}$ is the matroid obtained from $\mathcal{M}$ by **deletion** of s, that is, a matroid on $\mathsf{S} - \mathsf{s}$ with rank function $r|_{\mathsf{S}-\mathsf{s}}$, while $\boldsymbol{\mathcal{M}/\mathsf{s}}$ is the matroid obtained from $\mathcal{M}$ by **contraction** of s, that is, a matroid on $\mathsf{S} - \mathsf{s}$ with a rank function $r_{\mathcal{M}/\mathsf{s}}(\mathsf{Q}) = r(\mathsf{Q} \cup \mathsf{s}) - 1$. The **$k$-sum** of the matroid $\mathcal{M}$ is the matroid whose independent sets are those sets that can be partitioned into k independent sets of $\mathcal{M}$. For matroids $\mathcal{M}_1$ and $\mathcal{M}_2$ on disjoint sets S_1 and S_2 with rank functions r_1 and r_2, their **direct sum $\boldsymbol{\mathcal{M}_1 \oplus \mathcal{M}_2}$** is the matroid on $\mathsf{S}_1 \cup \mathsf{S}_2$ with rank function $r_\oplus(\mathsf{Q}) = r_1(\mathsf{Q} \cap \mathsf{S}_1) + r_2(\mathsf{Q} \cap \mathsf{S}_2)$ for all $\mathsf{Q} \subseteq \mathsf{S}_1 \cup \mathsf{S}_2$. Note that s is a bridge in $\mathcal{M}$ if and only if $\mathcal{M} \simeq (\mathcal{M} - s) \oplus U_{1,1}$.

3 Results

The first result on packing arborescences is due to Edmonds [5].

Theorem 1 ([5]) *Let $D = (V + s, A)$ be a rooted digraph and k a positive integer. There exists a packing of k spanning s-arborescences in D if and only if*

$$|\partial(X)| \geq k \text{ for all } \emptyset \neq X \subseteq V. \tag{4}$$

Edmonds [4] proved a much more general result on the intersection of two arbitrary matroids.

Theorem 2 ([4]) *Let $\mathcal{M}_1 = (\mathsf{S}, r_1)$ and $\mathcal{M}_2 = (\mathsf{S}, r_2)$ be two matroids and k a positive integer. There exists a common independent set of $\mathcal{M}_1$ and $\mathcal{M}_2$ of size k if and only if*

$$r_1(\mathsf{X}) + r_2(\mathsf{S} - \mathsf{X}) \geq k \text{ for all } \mathsf{X} \subseteq \mathsf{S}. \tag{5}$$

For matroids $\mathcal{M}_1$ and $\mathcal{M}_2$ on the same set S, one can find in polynomial time a maximum cardinality common independent set by the matroid intersection algorithm of Edmonds [4].

Theorem 1 was generalized in many directions. First, we mention the following result that can be proved by Theorem 2.

Theorem 3 ([2, 9]) *Let $D = (V + s, A)$ be a rooted digraph, k a positive integer and $\mathcal{M}_2 = (A, r_2)$ a matroid which is the direct sum of the matroids $\mathcal{M}_v = (\partial(v), r_v)$ for $v \in V$. There exists an $\mathcal{M}_2$-restricted packing of spanning s-arborescences in D if and only if*

$$r_2(\partial(X)) \geq k \text{ for all } \emptyset \neq X \subseteq V. \tag{6}$$

Durand de Gevigney, Nguyen and Szigeti [3] proved the following extension of Theorem 1.

Theorem 4 ([3]) *Let $D = (V + s, A)$ be a rooted digraph and $\mathcal{M}_1 = (\partial(V), r_1)$ a matroid. There exists an $\mathcal{M}_1$-based packing of s-arborescences in D if and only if*

$$r_1(\partial_s(X)) + |\partial_{V-X}(X)| \geq r_1(\partial(V)) \text{ for all } \emptyset \neq X \subseteq V. \tag{7}$$

In [14], the first author generalized Theorem 4 as follows.

Theorem 5 ([14]) *Let $D = (V + s, A)$ be a rooted digraph and $\mathcal{M}_1 = (\partial(V), r_1)$ a matroid. There exists an $\mathcal{M}_1$-reachability-based packing of s-arborescences in D if and only if*

$$r_1(\partial_s(X)) + |\partial_{V-X}(X)| \geq r_1(\partial_s(P(X))) \text{ for all } X \subseteq V. \tag{8}$$

In this paper, we prove the following result that is a common generalization of all the results previously mentioned in this paper.

Theorem 6 *Let $D = (V + s, A)$ be a rooted digraph, $\mathcal{M}_1 = (\partial(V), r_1)$ and $\mathcal{M}_2 = (A, r_2)$ two matroids such that $\mathcal{M}_2$ is the direct sum of the matroids $\mathcal{M}_v = (\partial(v), r_v)$ for $v \in V$. There exists an $\mathcal{M}_1$-reachability-based $\mathcal{M}_2$-restricted packing of s-arborescences in D if and only if*

$$r_1(F) + r_2(\partial(X) - F) \geq r_1(\partial_s(P(X))) \text{ for all } X \subseteq V \text{ and } F \subseteq \partial_s(X). \tag{9}$$

When we require $\mathcal{M}_1$-based packings, (9) can be simplified as follows.

Corollary 7 *Let $D = (V + s, A)$ be a rooted digraph, $\mathcal{M}_1 = (\partial(V), r_1)$ and $\mathcal{M}_2 = (A, r_2)$ two matroids such that $\mathcal{M}_2$ is the direct sum of the matroids $\mathcal{M}_v = (\partial(v), r_v)$ for $v \in V$. There exists an $\mathcal{M}_1$-based $\mathcal{M}_2$-restricted packing of s-arborescences in D if and only if*

$$r_1(F) + r_2(\partial(X) - F) \geq r_1(\partial(V)) \text{ for all } \emptyset \neq X \subseteq V \text{ and } F \subseteq \partial_s(X). \tag{10}$$

It is proved in [8] that the problem of matroid-based packing of *spanning* arborescences is NP-complete, however, (7) is a necessary and sufficient condition for the case of several fundamental matroid classes, as follows.

Theorem 8 ([8]) *Let $D = (V + s, A)$ be a rooted digraph and $\mathcal{M}_1 = (\partial(V), r_1)$ a matroid of rank 2 or a graphic matroid or a transversal matroid. There exists an $\mathcal{M}_1$-based packing of spanning s-arborescences in D if and only if* (7) *holds.*

Observe that the arc set A' of an $\mathcal{M}_1$-based $\mathcal{M}_2$-restricted packing of s-arborescences is independent in $\mathcal{M}_2$ hence restricting $\mathcal{M}_2$ to A' we get the free matroid. Moreover, as an $\mathcal{M}_1$-based $\mathcal{M}_2$-restricted packing of s-arborescences is obviously $\mathcal{M}_1$-based, (7) also holds for $(V + s, A')$. Hence we get the following corollary from Corollary 7 and Theorem 8.

Corollary 9 *Let $D = (V + s, A)$ be a rooted digraph, $\mathcal{M}_1 = (\partial(V), r_1)$ a matroid of rank 2, a graphic matroid, or a transversal matroid, and $\mathcal{M}_2 = (A, r_2)$ a matroid that is the direct sum of matroids $\mathcal{M}_v = (\partial(v), r_v)$ for $v \in V$. There exists an $\mathcal{M}_1$-based $\mathcal{M}_2$-restricted packing of spanning s-arborescences in D if and only if* (10) *holds.*

4 Preliminaries

Before proving Theorem 6, we provide some lemmas that will be useful later. Let D, $\mathcal{M}_1$ and $\mathcal{M}_2$ be as in Theorem 6. For $X \subseteq V$ and $F \subseteq \partial_s(X)$, let

$$\begin{aligned} \boldsymbol{b(X, F)} &:= r_1(F) + r_2(\partial(X) - F), \\ \boldsymbol{p(X)} &:= r_1(\partial_s(P(X))). \end{aligned}$$

The submodularity of b was proved in [2]. However, we need a bit more hence we give the full proof of the following lemma.

Lemma 10 *Let $X, X' \subseteq V$, $F \subseteq \partial_s(X)$ and $F' \subseteq \partial_s(X')$. Then*

$$b(X, F) + b(X', F') \geq b(X \cap X', F \cap F') + b(X \cup X', F \cup F'). \tag{11}$$

Moreover, if $e \in (\partial(X) - F) - (\partial(X') - F')$, then

$$r_1(F) + r_1(F') + r_2(\partial(X) - (F \cup e)) + r_2((\partial(X') - F') \cup e) \geq b(X \cap X', F \cap F') + b(X \cup X', F \cup F'). \tag{12}$$

Proof: First note that

$$\begin{aligned} (\partial_{\overline{X}}(x) - F) \cap (\partial_{\overline{X'}}(x) - F') &\supseteq \partial_{\overline{X \cup X'}}(x) - (F \cup F') \quad \text{for every } x \in X \cup X', \text{ and} & (13) \\ (\partial_{\overline{X}}(x) - F) \cup (\partial_{\overline{X'}}(x) - F') &\supseteq \partial_{\overline{X \cap X'}}(x) - (F \cap F') \quad \text{for every } x \in X \cap X'. & (14) \end{aligned}$$

By $\mathcal{M}_2 = \bigoplus_{v \in V} \mathcal{M}_v$, the monotonicity and the submodularity of r_2, (13) and (14), we get

$$\begin{aligned} & r_2(\partial(X) - F) + r_2(\partial(X') - F') \\ = & \sum_{x \in X} r_x(\partial_{\overline{X}}(x) - F) + \sum_{x \in X'} r_x(\partial_{\overline{X'}}(x) - F') \\ = & \sum_{x \in X - X'} r_x(\partial_{\overline{X}}(x) - F) + \sum_{x \in X' - X} r_x(\partial_{\overline{X'}}(x) - F') \\ & + \sum_{x \in X \cap X'} \left(r_x(\partial_{\overline{X}}(x) - F) + r_x(\partial_{\overline{X'}}(x) - F') \right) \\ \geq & \sum_{x \in (X - X') \cup (X' - X) \cup (X \cap X')} r_x(\partial_{\overline{X \cup X'}}(x) - (F \cup F')) + \sum_{x \in X \cap X'} r_x(\partial_{\overline{X \cap X'}}(x) - (F \cap F')) \\ = & r_2(\partial(X \cup X') - (F \cup F')) + r_2(\partial(X \cap X') - (F \cap F')). \end{aligned}$$

We get (11) by the above inequality and by the submodularity of r_1.
Note that, by $e = uv \in (\partial(X) - F) - (\partial(X') - F')$,

$$\begin{aligned} (\partial_{\overline{X}}(v) - (F \cup e)) \cap ((\partial_{\overline{X'}}(v) - F') \cup e) &= (\partial_{\overline{X}}(v) - F) \cap (\partial_{\overline{X'}}(v) - F'), & (15) \\ (\partial_{\overline{X}}(v) - (F \cup e)) \cup ((\partial_{\overline{X'}}(v) - F') \cup e) &= (\partial_{\overline{X}}(v) - F) \cup (\partial_{\overline{X'}}(v) - F'). & (16) \end{aligned}$$

By (15), (16) and the previous proof, (12) follows. □

Although $p(X)$ is not supermodular in general, we will prove the supermodular inequality for specific pairs in the next lemma, following an idea from [14].

Lemma 11 *Let X and X' be two subsets of V and $v \in X \cap X'$ such that $X' \subseteq P(v)$. Then*

$$p(X) + p(X') \leq p(X \cap X') + p(X \cup X'). \tag{17}$$

Proof: Since the reachability is transitive and $v \in X \cap X'$, we get $P(X') \subseteq P(X \cap X')$ and hence $\partial_s(P(X')) \subseteq \partial_s(P(X \cap X'))$. Furthermore, $P(X) \subseteq P(X \cup X')$ is obvious hence $\partial_s(P(X)) \subseteq \partial_s(P(X \cup X'))$. Thus, by the monotonicity of the rank function r_1, we get (17). □

5 Proof of Theorem 6

Observe that the existence of an $\mathcal{M}_1$-reachability-based $\mathcal{M}_2$-restricted packing of s-arborescences and that of s-one-arborescences are equivalent as an s-arborescence can be split into multiple s-one-arborescences. Hence, in the following proof, we will use s-one-arborescences.

Necessity: Let $\{T_1, \ldots, T_t\}$ be an $\mathcal{M}_1$-reachability-based $\mathcal{M}_2$-restricted packing of s-one-arborescences in D. As each T_i is an s-*one*-arborescence, $\partial(V) \cap A(T_i) = \partial(V) \cap A(T_i[s,v])$ for every $v \in V(T_i)$. For every vertex $v \in V$, let $\mathsf{B}_v = \{e_i = \partial(V) \cap A(T_i), v \in V(T_i)\}$. Let now $X \subseteq V, F \subseteq \partial_s(X)$ and $\mathsf{B} = \bigcup_{v \in X} \mathsf{B}_v$. Since $\mathrm{Span}_{\mathcal{M}_1}$ is monotone, by (3) and definition of $P(X)$, we have $\mathrm{Span}_{\mathcal{M}_1}(\mathsf{B}) \supseteq \bigcup_{v \in X} \mathrm{Span}_{\mathcal{M}_1}(\mathsf{B}_v) \supseteq \bigcup_{v \in X} \partial_s(P(v)) = \partial_s(P(X))$. Then, since r_1 is monotone, we have the following inequality $(*)$ $r_1(\mathsf{B}) \geq r_1(\partial_s(P(X)))$. For each $e_i \in \mathsf{B} - F$, there exists a vertex $v \in X$ such that $e_i \in \mathsf{B}_v$ and then since T_i is an s-arborescence and $v \in V(T_i) \cap X$, there exists $a_i \in A(T_i) \cap (\partial(X) - F)$. Since r_2 is monotone, $\{a_i : e_i \in \mathsf{B} - F\}$ is independent in $\mathcal{M}_2$, these arborescences are edge-disjoint, r_1 is subcardinal, submodular and monotone and by $(*)$, we have $r_2(\partial(X) - F) \geq r_2(\{a_i : e_i \in \mathsf{B} - F\}) = |\{a_i : e_i \in \mathsf{B} - F\}| = |\mathsf{B} - F| \geq r_1(\mathsf{B} - F) \geq r_1(\mathsf{B} \cup F) - r_1(F) \geq r_1(\mathsf{B}) - r_1(F) \geq r_1(\partial_s(P(X))) - r_1(F)$ that is, (9) is satisfied.

Sufficiency: We suppose that the theorem is not true. Let us take a counterexample $(D, \mathcal{M}_1, \mathcal{M}_2)$ ((9) is satisfied but the desired packing of s-*one*-arborescences does not exist) that first minimizes the number of arcs in D and then the number of non-bridge edges in $\mathcal{M}_2$.

We say that a pair consisting of $X \subseteq V$ and $F \subseteq \partial_s(X)$ is **tight** if $b(X,F) = p(X)$ and is **critical for an edge** e if (X,F) is tight and e is a bridge in $\mathcal{M}_2|_{\partial(X)-F}$.

Case 1. First suppose that there exists an edge e for which no critical pair exists. Then the following hold.

$$r_1(F) + r_2(\partial^{D-e}(X) - F) \geq r_1(\partial_s^{D-e}(P_{D-e}(X))) \text{ for all } X \subseteq V \text{ and } F \subseteq \partial_s^{D-e}(X), \quad (18)$$

$$r_1(\partial_s^{D-e}(P_{D-e}(w))) = r_1(\partial_s(P(w))) \text{ for every } w \in V. \quad (19)$$

Proof: First, suppose to the contrary that there exist $X \subseteq V$ and $F \subseteq \partial_r^{D-e}(X)$ that violates (18). Then, by (9), the subcardinality of r_2, (X,F) violates (18) and the monotonicity of r_1, we have $r_1(\partial_s(P(X))) \leq r_1(F) + r_2(\partial(X) - F) \leq r_1(F) + r_2(\partial^{D-e}(X) - F) + 1 \leq r_1(\partial_s^{D-e}(P_{D-e}(X))) \leq r_1(\partial_s(P(X)))$, so equality holds everywhere. Hence (X,F) is tight in D and e is a bridge in $\mathcal{M}_2|_{\partial(X)-F}$. Therefore, (X,F) is critical for e, a contradiction.

Now, suppose to the contrary that there exists $w \in V$ that violates (19). Let $F' = \partial_s^{D-e}(P_{D-e}(w)), F = \partial_s(P(w))$ and $X' = P_{D-e}(w)$. By the definition of X' and F', $\partial(X') - F' \subseteq \{e\}$. Then, by w violates (19), $F' \subseteq F, P(w) \subseteq P(X')$, the monotonicity of r_1, (9) and the subcardinality of r_2, we have $r_1(F') + 1 \leq r_1(F) = r_1(\partial_s(P(w))) \leq r_1(\partial_s(P(X'))) \leq r_1(F') + r_2(\partial(X') - F') \leq r_1(F') + 1$. Thus equality holds everywhere. Hence (X', F') is tight, and $r_2(\partial(X') - F') = r_2(e) = 1$, that is, e is a bridge in $\mathcal{M}_2|_{\partial(X')-F'}$. Therefore, (X', F') is critical for e, a contradiction. □

By (18), $(D - e, \mathcal{M}_1 - e, \mathcal{M}_2 - e)$ satisfies the condition of the theorem. By $|A(D - e)| < |A(D)|$, it is not a counterexample, so there exists an $(\mathcal{M}_1 - e)$-reachability-based $(\mathcal{M}_2 - e)$-restricted packing $T_1, \ldots, T_t$ of s-one-arborescences in $D - e$, that is, for every $v \in V$, $\{A(T_i) \cap \partial^{D-e}(v) : v \in V(T_i)\}$ is independent in $\mathcal{M}_v - e$ (and hence in $\mathcal{M}_v$) and $\{A(T_i) \cap \partial^{D-e}(V) : v \in V(T_i)\}$ is independent in $\mathcal{M}_1 - e$ (and hence in $\mathcal{M}_1$) of size $r_1(\partial_r^{D-e}(P_{D-e}(v)))$ that is, by (19), of size $r_1(\partial_s(P(v)))$. Then $T_1, \ldots, T_t$ is an $\mathcal{M}_1$-reachability-based $\mathcal{M}_2$-restricted packing of s-one-arborescences in D, and the proof is complete in this case.

Case 2. Suppose now that there exists a non-bridge edge $e = uv$ in $\mathcal{M}_2$. Since we are not in Case 1, there exists a critical pair (X,F) for e such that X is minimal.

Claim 12 $X \subseteq P(v)$.

PROOF: Let $(X', F') = (P(v), \partial_s(P(v)))$. By $\partial(X') - F' = \emptyset$ and $\partial_s(P(X')) = F'$, we get $r_2(\partial(X') - F') = r_2(\emptyset) = 0 = r_1(\partial_s(P(X'))) - r_1(F')$, that is (X', F') is tight. By the tightness of (X, F) and (X', F'), Lemma 11, (9) and (11), we have $b(X, F) + b(X', F') = p(X) + p(X') \leq p(X \cap X') + p(X \cup X') \leq b(X \cap X', F \cap F') + b(X \cup X', F \cup F') \leq b(X, F) + b(X', F')$. Hence equality holds everywhere, in particular, $(X \cap X', F \cap F')$ is tight. Note that, by $X' = P(v)$ and $uv \in \partial(X) - F$, $e \in Y := \partial(X \cap X') - (F \cap F')$. Suppose that e is not a bridge in $\mathcal{M}_2|_Y$. Then there exists an $\mathcal{M}_2$-base B' of Y not containing e. Since no edge exists from $X - X'$ to $X \cap X'$, $B' \subseteq \partial(X) - F$ so there exists an $\mathcal{M}_2$-base B of $\partial(X) - F$ containing B'. Since B' was an $\mathcal{M}_2$-base of Y, $e \notin B$. Thus e is not a bridge in $\mathcal{M}_2|_{\partial(X)-F}$, which is a contradiction. So e is a bridge in $\mathcal{M}_2|_{\partial(X \cap X') - (F \cap F')}$, thus $(X \cap X', F \cap F')$ is a critical pair for e. It follows, by the minimality of X, that $X \subseteq X' = P(v)$. □

Let $\mathcal{M}_2' = (\mathcal{M}_2/e) \oplus e$ (with rank function r_2'), that is, $\mathcal{M}_2'$ is obtained from $\mathcal{M}_2$ by contracting e and adding back e as a bridge. Note that $\mathcal{M}_2'$ will still be a direct sum of its submatroids on $\partial(w)$ for $w \in V$ as $\mathcal{M}_2' = \bigoplus_{w \in V - v} \mathcal{M}_w \oplus \mathcal{M}_v'$ where $\mathcal{M}_v' = (\mathcal{M}_v/e) \oplus e$. We show now that (9) with respect to $\mathcal{M}_2'$ holds, that is,

$$b'(X', F') := r_1(F') + r_2'(\partial(X') - F') \geq r_1(\partial_s(P(X'))) \text{ for all } X' \subseteq V \text{ and } F' \subseteq \partial_s(X'). \tag{20}$$

PROOF: Assume for a contradiction that there exists (X', F') that violates (20), that is, $b'(X', F') \leq p(X') - 1$. By the definition of contraction, $r_2'(Y) = r_2(Y)$ if $e \in Y$ and $r_2(Y \cup e) - 1$ if $e \notin Y$. It follows, by (9) for (X', F') and the monotinicity of r_2, that $p(X') \leq b(X', F') \leq r_1(F') + r_2((\partial(X') - F') \cup e) \leq b'(X', F') + 1$. By adding the above two inequalities, we get that all these inequalities hold with equalities, so $v \in X', e \notin \partial(X') - F'$ and $r_1(F') + r_2((\partial(X') - F') \cup e) = p(X')$. Since (X, F) is a critical pair for e, $e \in \partial(X) - F$, $r_1(F) + r_2(\partial(X) - (F \cup e)) + 1 = b(X, F) = p(X)$ and, by Claim 12, we have $X \subseteq P(v)$ hence the condition of Lemma 11 is satisfied. By the two equalities above, Lemma 11, (9) for the pairs $(X \cap X', F \cap F')$ and $(X \cup X', F \cup F')$ and (12), we get a contradiction. □

By (20), $(D, \mathcal{M}_1, \mathcal{M}_2')$ satisfies the condition of the theorem. Note that if f is a bridge in $\mathcal{M}_2$, then it will be a bridge in $\mathcal{M}_2'$ also. Then the number of non-bridge edges in $\mathcal{M}_2'$ is one less than in $\mathcal{M}_2$, hence $(D, \mathcal{M}_1, \mathcal{M}_2')$ is not a counterexample, so there exists an $\mathcal{M}_1$-reachability-based $\mathcal{M}_2'$-restricted packing $T_1, \ldots, T_t$ of s-one-arborescences in D, that is, for every $v \in V$, $Y = \{A(T_i) \cap \partial(v) : v \in V(T_i)\}$ is independent in $\mathcal{M}_v'$ (and hence, by $r_v(Y) \leq |Y| = r_v'(Y) \leq r_v(Y)$, independent in $\mathcal{M}_v$) and $\{A(T_i) \cap \partial(V) : v \in V(T_i)\}$ is independent in $\mathcal{M}_1$ of size $r_1(\partial_s(P(v)))$. Then, as the independent sets of $\mathcal{M}_2'$ are also independent in $\mathcal{M}_2$, $T_1, \ldots, T_t$ is an $\mathcal{M}_1$-reachability-based $\mathcal{M}_2$-restricted packing of s-one-arborescences in D, and the proof is complete in this case.

Case 3. We may suppose finally that each edge is a bridge in $\mathcal{M}_2$, that is, $\mathcal{M}_2$ is the free matroid. Note that in this case (9) implies (8) and hence we can conclude by Theorem 5. □□

6 Algorithmic aspects

We show in this section how to derive from our proof a polynomial algorithm to find either a reachability-based matroid-restricted packing of s-one-arborescences or a pair (X, F) that violates (9).

First we show how to check in polynomial time whether (9) holds. We start with the following observation.

Lemma 13 *If there exists a pair that violates* (9)*, then there also exists a pair* (X^*, F^*) *violating* (9) *and a vertex* v *such that* $v \in X^* \subseteq P(v)$.

The proof will be similar to the proof of Claim 12.

PROOF: Let (X, F) be a pair that violates (9) such that X is maximal and F is also maximal with respect to X. Let $v \in X$. If $X \subseteq P(v)$, then $(X^*, F^*) := (X, F)$ is as required. Otherwise, let

$(X', F') = (P(v), \partial_s(P(v)))$. By $\partial(X') - F' = \emptyset$ and $\partial_s(P(X')) = F'$, we get $r_2(\partial(X') - F') = r_2(\emptyset) = 0 = r_1(\partial_s(P(X'))) - r_1(F')$, that is (X', F') is tight. By (11), as (X, F) violates (9) and by the tightness (X', F'), and by Lemma 11, we have $b(X \cap X', F \cap F') + b(X \cup X', F \cup F') \leq b(X, F) + b(X', F') < p(X) + p(X') \leq p(X \cap X') + p(X \cup X')$. Hence $(X \cap X', F \cap F')$ or $(X \cup X', F \cup F')$ is violating (9). By the maximality of (X, F), if $P(v) \not\subseteq X$ or $\partial_s(P(v)) \not\subseteq F$, then $(X \cup X', F \cup F')$ does not violate (9). Thus, in this case, $(X^*, F^*) = (X \cap X', F \cap F')$ is violating (9), moreover, by the definition of X', $v \in X^* \subseteq P(v)$ as required. Therefore, we find a violating pair as required except when $P(v) \subseteq X$ and $\partial_s(P(v)) \subseteq F$. However, this cannot hold for all $v \in X$ as then $X = \bigcup_{v \in X} P(v) = P(X)$ and $F = \partial_s(X) = \partial(X)$ hence (9) holds with equality, a contradiction. □

By Lemma 13, (9) holds if and only if for every $v \in V$, it holds for all pairs (X, F) with the addition property that $v \in X \subseteq P(v)$. Note that for a fixed vertex v, for all $v \in X \subseteq P(v)$, $P(X) = P(v)$, so the right hand side of (9) is constant.

On the one hand, for a fixed set $X \subseteq V$, $r_1(F) + r_2(\partial(X) - F)$ for all $F \subseteq \partial_s(X)$ is a submodular function, so by submodular function minimization one can determine in polynomial time, for all $X \subseteq V$, $q(X) = \min\{r_1(F) + r_2(\partial(X) - F) : F \subseteq \partial_s(X)\}$. On the other hand, by Lemma 10, $q(X)$ is submodular. Then, using again submodular function minimization, one can check in polynomial time whether for a fixed $v \in V$, for all $v \in X \subseteq P(v)$, $q(X) \geq r_1(\partial_s(P(v)))$. We may hence conclude that we can check in polynomial time whether (9) holds.

It follows that (8) can also be checked in polynomial time. Then the proof of Theorem 5 in [14] provides a polynomial algorithm to find either a Reachability-based packing of s-one-arborescences or a set that violates (8).

Now we can explain our algorithm. We check first whether (9) holds. As mentioned above, in polynomial time, either we find a set that violates (9) and we stop or we know that (9) holds and we continue. If every edge is a bridge in $\mathcal{M}_2$ then the problem reduces to the problem of reachability-based packing of s-one-arborescences and hence we are done by the above remark on the algorithm of [14]. If there exists a non-bridge edge in $\mathcal{M}_2$, then let us choose one, say e. Let us check if $(D - e, \mathcal{M}_1 - e, \mathcal{M}_2 - e)$ satisfies (18) and (19). (18) is just (9) for the smaller graph, so we can do it. The second one is trivially polynomial to check. If both hold, then recursively we use our algorithm for $(D - e, \mathcal{M}_1 - e, \mathcal{M}_2 - e)$ and the packing obtained will be a required packing for $(D, \mathcal{M}_1, \mathcal{M}_2)$. Otherwise, $(D, \mathcal{M}_1, \mathcal{M}_2')$, where $\mathcal{M}_2'$ is defined in Case 2 in the proof of Theorem 6, satisfies (20) and recursively we use our algorithm for $(D, \mathcal{M}_1, \mathcal{M}_2')$ and the packing obtained will be a required packing for $(D, \mathcal{M}_1, \mathcal{M}_2)$. Note that during the recursive execution of our algorithm either the number of edges decreases by one or the number of non-bridge edges in $\mathcal{M}_2$ decreases by one, hence our algorithm stops in polynomial time.

The above argument shows that the following theorem holds.

Theorem 14 *Let $D = (V + s, A)$ be a rooted digraph, $\mathcal{M}_1 = (\partial(V), r_1)$ and $\mathcal{M}_2 = (A, r_2)$ two matroids such that $\mathcal{M}_2$ is the direct sum of the matroids $\mathcal{M}_v = (\partial(v), r_v)$ for $v \in V$. There exists a polynomial algorithm to find either an $\mathcal{M}_1$-reachability-based $\mathcal{M}_2$-restricted packing of s-arborescences in D or a pair (X, F) that violates (9).* □

7 Concluding remarks

7.1 An extension for dypergraphs

A **dypergraph** is a directed hypergraph where each oriented hyperedge, called a **dyperedge**, has one head and multiple tails. An **s-hyperarborescence** is a dypergraph which can be trimmed to an s-arborescence, that is, each of its dyperedges can be substituted by one arc from one of its tails to its head such that the resulting digraph is an s-arborescence. [7] showed that all arborescence packing results can be simply generalized to dypergraphs. The idea is to substitute each dyperedge of the input dypergraph by a new vertex such that it is entered by multiple arcs from each of the tails of the dyperedge and it has only one outgoing arc, called a head arc, that has the same head as the dyperedge. By the same

construction one can get a generalization of the result presented here, one only needs to add the free matroid on $\partial(v)$ for each new vertex v and keep the original matroid $\mathcal{M}_2$ on the head arcs.

7.2 Open problems

We conclude this paper with some remarks on the weighted versions of the problems. Suppose that we are given a weight function on the set of arcs of a digraph. The weight of a packing of arborescences is the sum of the weights of the arcs of the arborescences in the packing. It is clear that one can find a packing of k spanning s-arborescences of minimum weight (if one exists) with the weighted matroid intersection algorithm [6]. Similarly, a matroid-restricted packing of spanning s-arborescences of minimum weight (if one exists) can be found with the weighted matroid intersection algorithm. The weighted version of the problem of matroid-based packing of s-arborescences was solved in [3] by the ellipsoid method [11] and submodular function minimization [12, 15]. It is an open problem whether there exists a polynomial algorithm to solve the common generalization of these problems, that is to find a matroid-based matroid-restricted packing of s-arborescences of minimum weight.

Finally, we note that the weighted version of the problem of reachability-based packing of s-arborescences was solved in [1] by an abstract reformulation of the problem. Obviously, the problem of reachability-based matroid-restricted packing of s-arborescences of minimum weight also remains open.

Acknowledgments

Research was supported by the Project RIME of the laboratory G-SCOP. The first author was also supported by the Hungarian Scientific Research Fund – OTKA, K109240, and by the MTA-ELTE Egerváry Research Group.

References

[1] K. Bérczi, T. Király, Y. Kobayashi: Covering intersecting bi-set families under matroid constraints. *SIAM J. Discrete Math.*, 30-3, 1758–1774 (2016)

[2] A. Bernáth, T. Király, Blocking optimal k-arborescences, in: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, SIAM, Philadelphia, PA, USA, 2016, pp. 1682–1694.

[3] O. Durand de Gevigney, V. H. Nguyen, Z. Szigeti, Matroid-Based Packing of Arborescences, *SIAM J. Discrete Math.*, 27 (2013), pp. 567–574.

[4] J. Edmonds, Submodular functions, matroids, and certain polyhedra, in: *Combinatorial Structures and their Applications*, R. Guy, H. Hanani, N. Sauer, and J. Schönheim eds., Gordon and Breach, New York, 1970, pp. 69–87.

[5] J. Edmonds, Edge-disjoint branchings, in *Combinatorial Algorithms*, B. Rustin ed., Academic Press, New York, 1973, pp. 91–96.

[6] J. Edmonds, Matroid intersection, *Ann. Disc. Math.*, 4 (1979) 29–49.

[7] Q. Fortier, Cs. Király, M. Léonard, Z. Szigeti, A. Talon: Old and new results on packing arborescences. *EGRES Technical Report* No. TR-2016-04, `wwww.cs.elte.hu/egres` (2016)

[8] Q. Fortier, Cs. Király, Z. Szigeti, S. Tanigawa, On packing spanning arborescences with matroid constraint, *EGRES Technical Report* No. TR-2016-18, `www.cs.elte.hu/egres`, 2016.

[9] A. Frank, Rooted k-connections in digraphs, *Discrete Applied Mathematics* 157 (2009) 1242-1254.

[10] A. Frank, Connections in Combinatorial Optimization, *Oxford University Press*, 2011.

[11] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica*, Springer Berlin, 1 (2) (1981) 169-197.

[12] S. Iwata, L. Fleischer, S. Fujishige, A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM* 48, 4 (2001) 761-777.

[13] N. Kamiyama, N. Katoh, A. Takizawa, Arc-disjoint in-trees in directed graphs, *Combinatorica*, 29 (2009) 197-214.

[14] Cs. Király, On maximal independent arborescence packing, *SIAM Journal on Disc. Math.*, 30(4) (2016) 2107–2114.

[15] A. Schrijver, A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *J. Combin. Theory, Ser. B* 80 (2000) 346-355.

Finding strongly popular matchings in certain bipartite preference systems

Tamás Király[1]

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
tkiraly@cs.elte.hu

Zsuzsa Mészáros-Karkus[1]

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
karkuszsuzsi@gmail.com

Abstract: The computational complexity of the bipartite popular matching problem depends on whether ties are allowed in the preference lists. If one side has strict preferences while nodes on the other side are indifferent (but prefer to be matched), then a popular matching can be found in polynomial time [Cseh, Huang, Kavitha, 2015]. However, as the same paper points out, the problem becomes NP-complete if one side has strict preferences while the other side can have both indifferent nodes and nodes with strict preferences. We show that the problem of finding a *strongly* popular matching is polynomial-time solvable even in the latter case.

Keywords: graph algorithms, stable marriage, popular matching

1 Introduction

A *bipartite preference system with ties* consists of a bipartite multigraph $G = (S, T; E)$ and partial orders $\preceq_v$ on the edges incident to v, for every node $v \in S \cup T$. Given a bipartite preference system with ties, a node *prefers* a matching M_1 to a matching M_2 if it is either matched in M_1 but not in M_2, or matched by a better edge in M_1 than in M_2. Matching M_1 is *more popular* than matching M_2 if the number of nodes preferring M_1 to M_2 is strictly larger than the number of nodes preferring M_2 to M_1. This relation is not transitive; it is possible that M_1 is more popular than M_2, M_2 is more popular than M_3, and M_3 is more popular than M_1 [2]. A matching M is *popular* if no matching is more popular than M, and it is *strongly popular* if M is more popular than any other matching. These notions were first introduced by Gärdenfors [8], who showed that *a)* every strongly popular matching is stable and *b)* in case of no ties, all stable matchings are popular.

Obviously, an instance cannot have two strongly popular matchings, because both of them would be more popular than the other, which is impossible. Furthermore, a strongly popular matching must be a unique popular matching. However, there are instances where the popular matching is unique but it is not strongly popular; see the full version of [2] for an example.

Algorithmic questions about popular matchings have generated a lot of interest lately, see Section 1.1 for a short summary of recent results. Here we just mention that for any preference system with ties (even non-bipartite), it can be decided in polynomial time if a given matching is popular or strongly popular [2]. This means that the decision problem for popular matchings is in the complexity class NP, while the decision problem for strongly popular matchings is in the lesser-known complexity class UP (Unambiguous Polynomial-time). The latter class, introduced by Valiant [15], consists of the decision problems solvable by an NP machine such that all witnesses are rejected in a "no" instance, while exactly one witness is accepted in a "yes" instance. The strongly popular matching problem belongs to this class because each "yes" instance has a single strongly popular matching and it can be verified in polynomial time.

[1]Research is supported by the Hungarian National Research, Development and Innovation Office – NKFIH, grant number K120254. The authors are members of the MTA-ELTE Egerváry Research Group.

In this paper, we consider bipartite preference systems with two types of nodes: *nodes with strict preferences*, where the preference order $\preceq_v$ is a total order on $\delta_G(v)$, and *indifferent nodes*, where every incident edge is equally good (but who still prefer to be matched). If all nodes have strict preferences, then every stable matching is popular [8]. On one hand, this implies that there always exists a popular matching and one can be found using the well-known Gale-Shapley algorithm [7]. On the other hand, we can decide if a strongly popular matching exists by finding an arbitrary stable matching and checking whether it is strongly popular (this also works in non-bipartite preference systems without ties [2]).

The problems become more complex when indifferent nodes are also allowed on one of the sides. If nodes on one side have strict preferences while those on the other side are all indifferent, then the existence of a popular matching can still be decided in polynomial time, as shown by Cseh, Huang, and Kavitha [4]. However, they also showed that the problem becomes NP-complete if one side has strict preferences while the other side may feature both indifferent nodes and nodes with strict preferences; see the full version of [4] and [5] for proofs.

The main result of the present paper is that the existence of a *strongly* popular matching can be decided in polynomial time even in the latter case.

Theorem 1 *Given a bipartite preference system $(G = (S, T; E), \preceq)$ where nodes in S have strict preferences and each node in T is either indifferent or has strict preferences, it can be decided in polynomial time if there is a strongly popular matching.*

The algorithm successively finds edges that *cannot* be in a strongly popular matching or *must* be in any strongly popular matching, and also maintains a directed graph related to the possible structure of the strongly popular matching. The set of possible candidates keeps shrinking until, at the end, we can either conclude that there is no strongly popular matching, or exactly one candidate matching remains. In the latter case, we can check in polynomial time whether this matching is strongly popular or not.

1.1 Other related work

There is a lot of ongoing research about the computational complexity of the popular matching problem. For bipartite preference systems with no ties, Huang and Kavitha [9] showed that a maximum size popular matching can be found in polynomial time, and Cseh and Kavitha [6] gave an algorithm for deciding if a given edge belongs to a popular matching. The former result can also be extended to the Hospitals-Residents problem, where more than one residents can be matched to a hospital [3, 14]. On the other hand, the complexity of deciding the existence of a popular matching in a non-bipartite preference system without ties is still open. Huang and Kavitha [10] introduced the notion of unpopularity factor, and showed that, for any positive ε, it is NP-hard to compute a matching with unpopularity factor within $\frac{4}{3} - \varepsilon$ of optimal.

Several recent results concern a slightly different, one-sided model (also called the House Allocation model), where one side has preference lists, while nodes on the other side do not vote at all and do not prefer to be matched. Abraham et al. [1] gave a polynomial-time algorithm for finding a popular matching in this model. If the preferences are strict, then optimal popular matchings can also be found for various notions of optimality [12, 13].

2 Proof of the main theorem

In this section we prove Theorem 1. We are given a bipartite multigraph $G = (S, T; E)$, and the node set T is partitioned into two parts, T_P and T_I. The nodes in $S \cup T_P$ have strict preference orders $\preceq_v$ over their incident edges, while the nodes in T_I are indifferent but prefer to be matched. We give a polynomial-time algorithm which decides if the instance admits a strongly popular matching (SPM for short).

2.1 Preliminaries

Before going into the details, we give an overview of the main ideas of the proof. During the algorithm, we modify the instance using the following two operations.

1. We remove edges that cannot appear in an SPM of the current instance,
2. We fix edges that must belong to the SPM of the current instance (if it exists). Fixed edges are removed together with their two endnodes. The set of fixed edges is denoted by F.

Let $G^k = (S^k, T^k; E^k)$ be the current instance after performing k of the above operations, and let F be the set of edges fixed so far.

Lemma 2 *If the original instance has an SPM M, then $F \subseteq M$, and $M \setminus F$ is an SPM of G^k.*

Proof: We prove by induction on k; let G^{k-1} be the instance before the last operation. If the last operation was the removal of an edge st, then, by induction, M contains F, $M \setminus F$ is an SPM of G^{k-1}, and $st \notin M$. Thus $M \setminus F$ is an SPM of G^k.

If we fixed an edge st in the last operation, then $M \setminus (F - st)$ is an SPM of G^{k-1} by induction, and $st \in M \setminus (F - st)$ because we only fix edges with this property. This implies that $st \in M$, and therefore $M \setminus F$ is an SPM of G^k. □

Note that it is possible that G^k has an SPM even if G does not have one. However, this is not a problem: if we eventually obtain an empty graph by repeating the operations, then F is the only candidate for an SPM, and we can check in polynomial time if it is an SPM of G or not. On the other hand, if we obtain a graph G^k that has no SPM, then G also has none.

An edge $st \in E$ is called a *blocking edge* with respect to a matching M if both s and t prefer the edge st to their partner in the matching (this includes the case when $t \in T_I$ and it is unmatched). If M is an SPM, then there is no blocking edge with respect to M; indeed, if M' is the matching obtained from M by adding a blocking edge st and removing the original edges incident to s and t, then M is not more popular than M'. We will use the term "blocking edge" in another sense for parallel edges: if e and e' are parallel edges and one endpoint prefers e to e', then e blocks any matching M containing e', since $M - e' + e$ is at least as popular as M.

In addition to blocking edges, we will use alternating paths and cycles to show that certain matchings cannot be strongly popular. Given a matching M and an alternating path or cycle w.r.t. M, let M' be the matching obtained from M by exchanging along the path or cycle (if we exchange along a path whose first or last edge is not in M, then we also remove the edge of M covering the corresponding endpoint of the path). If we can show that M' is at least as popular as M, then M is not an SPM.

2.2 First phase of the algorithm

The algorithm starts with a first phase that is reminiscent of the first phase of Irving's algorithm for the stable roommates problem [11]. We repeat the following steps.

- From every node in $S \cup T_P$ we draw a directed edge to its first choice.
- If there is a directed edge to a node $v \in S \cup T_P$, then we delete the edges incident to v which are worse according to $\preceq_v$ than the incoming directed edge. We also delete the edges parallel to the directed edge. If we deleted a node's first choice, then we draw a directed edge to its first choice among its remaining neighbors.

Claim 3 *The deleted edges cannot belong to an SPM.*

Proof: Suppose that uv belongs to an SPM and it was deleted because of a directed edge wv. Then wv is a blocking edge with respect to the SPM, a contradiction. □

Claim 4 *If at some point there is only one directed edge st entering a node in T_I, then st belongs to the SPM if there is one.*

PROOF: Suppose that the SPM M does not contain st; then t has to be matched to a node $u \neq s$, otherwise st would be blocking. Consider the path that starts with s and alternates between directed edges and edges of M. (The first edge is st, the second is tu.) If we reach a node $t' \in T_I$, then by exchanging along the path we get a matching which is as popular as M: the nodes of S on the path all get a better partner, while the only nodes that may prefer M are the nodes of T in the path except for t and t', and the partners of t' and s in M. This contradicts the assumption that M is an SPM.

If we return to s, then exchanging along the cycle yields a matching at least as popular as M. See Figure 1 for an illustration of both cases. □

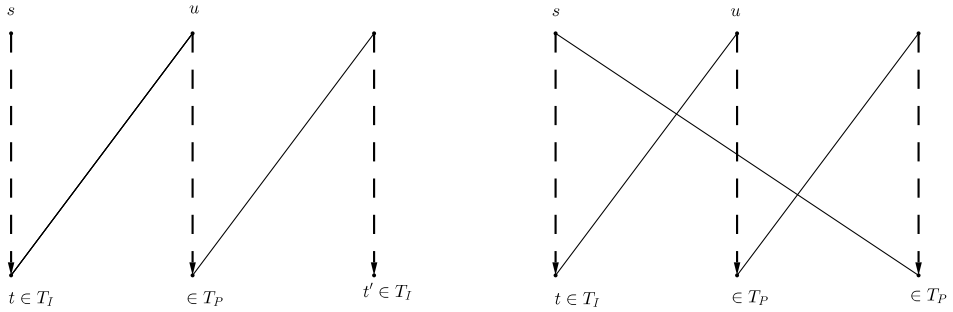


Figure 1: The solid edges belong to the SPM.

Claim 4 means that we can fix the edge st to be in F, and delete s and t from the graph.

Claim 5 *If at some point there is a node $t \in T_I$ which is not an endpoint of a directed edge but there is an edge st which has not been deleted, then an edge su cannot belong to the SPM if s prefers st over su.*

PROOF: Suppose that su is in the SPM. The node t has to be matched to some node v, since otherwise st would be blocking. Consider the path which starts with us, st, tv and then alternates between directed edges and edges of the SPM. Similarly to the proof of Claim 4, if we reach a node in T_I, then exchanging the edges along the path yields a matching preferred by the same number of nodes as the original SPM, while if the path returns to u, then by exchanging along the cycle we get a new matching that is as popular as the SPM, a contradiction. □

It follows from the claim that we can delete such edges su, and continue phase 1. We can also delete the nodes in $S \cup T_P$ which become isolated. If the graph becomes empty at the end of phase 1, then we can check whether the set F of fixed edges is an SPM in the original graph G, so we are done by Lemma 2. Otherwise we proceed to phase 2, which is described below.

2.3 Second phase of the algorithm

Let D' denote the directed graph obtained at the end of phase 1, and let G' be the bipartite graph consisting of all nodes and edges that have not been deleted in the first phase. D' can have three types of components:

- directed cycles;
- in-arborescences with a root-node (sink) in T_I having in-degree at least 2. The other nodes of the arborescence are in $S \cup T_P$ and they have out-degree 1 and in-degree at most 1. Therefore, each arborescence consists of disjoint directed paths leading to the root-node; the first nodes of these paths are called *leaves*.

- isolated nodes that are in T_I (note that these nodes are not isolated in G').

Let T_1 denote the nodes in T_I which are root-nodes of one of the arborescences, and let T_2 denote the isolated nodes in D'.

Lemma 6 *If uv is an edge in $G'[S \cup T_P \cup T_1]$ and it is not a directed edge in any direction, then uv cannot belong to the SPM.*

PROOF: Suppose uv is in the SPM M. If $u \in T_1$, then there is a directed edge su in D', for some $s \neq v$. Consider the path starting with su, uv and then alternating between directed edges and edges of M. Similarly to the proof of Claim 4, we either reach a node in T_I or return to s, and exchanging along the obtained path or cycle yields a matching that is preferred by at least as many nodes as the number of nodes that prefer M.

Now consider the case where $u \in T_P$. Consider the path starting with vu and then alternating between directed edges and edges of M. If we return to v without reaching a node in T_1, then exchanging along the cycle yields a matching that is at least as popular as M. If we reach a node in T_1, then there is another directed edge pointing to this node, which we add to the path. Let this path be denoted by P. We continue P from v with edges alternating between directed edges and edges of M. If we reach a node in T_I, then exchanging along the path yields a matching that is at least as popular as M; see Figure 2 for an illustration of this case. If we return to a node in P, then, again, exchanging along the obtained cycle yields a matching at least as popular as M. (One of the endpoints of each edge in M is better off with the new matching except for maybe one edge, but u and v are both better off.) □

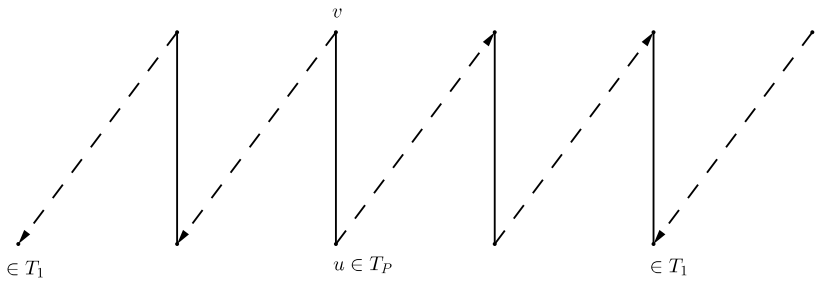


Figure 2: The solid edges belong to the SPM.

The lemma implies that only the edges of D' and edges of G' with one endpoint in T_2 can belong to the SPM. Therefore we can delete the other edges. From Claim 5, it follows that for every node $s \in S$ there can be only one edge between s and T_2.

Lemma 7 *If there is a cycle of length more than 2 in D' then there is no SPM.*

PROOF: Suppose that there is an SPM M. If one of the nodes v of the cycle is matched to a node $t \in T_2$ in M, then node v prefers its predecessor u in the cycle (because of Claim 5), and therefore uv is a blocking edge with respect to M.

If every node of the cycle is matched along the cycle, then we can exchange along the cycle to get a matching at least as popular as M. □

A cycle of length 2 in D' corresponds to a single edge that must belong to the SPM, so we can fix these edges and delete their endpoints.

Claim 8 *In the SPM, only the leaves (i.e. the nodes of in-degree 0 and out-degree 1 in D') can be matched to nodes in T_2.*

PROOF: Suppose u is matched to $t \in T_2$ in the SPM and u is not a leaf; therefore, there is a node v such that vu is in the arborescence. Because of Claim 5, vu is a blocking edge. □

By the claim, we can delete the edges between T_2 and any node which is not a leaf.

Claim 9 *Every leaf is matched in the SPM.*

PROOF: Let M be the SPM, and suppose there is a leaf $s \in S$ that is not matched in M. The other nodes of the branch containing s, except for the root, must be matched along the branch. By exchanging the edges along the branch such that the edge incident to s and the edge incident to the root belong to the new matching, we obtain a matching that is as popular as M.

Now suppose there is an unmatched leaf $t \in T$. Again, the other nodes of the branch must be matched along the branch, and now the root also has to be matched in this branch, otherwise there is a blocking edge. We exchange the edges along the branch and add to the matching another edge pointing to the root (here we use the property that the in-degree of the root is at least 2). If the tail of this edge is covered by M, then we remove the edge covering it from the matching. It is easy to check that the new matching is at least as popular as M. □

If there is an arborescence with all leaves in T, then all of its nodes have to be matched along the arborescence, and from the above claim all of its nodes have to be matched. But the arborescence has an odd number of nodes, therefore there cannot be an SPM.

If an arborescence has only one leaf in S, then its nodes have to be matched along the arborescence, and there is a unique way to match them (see Figure 3). Therefore we can fix these edges and delete the arborescence.

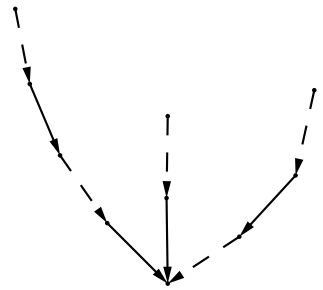


Figure 3: The dashed edges give the only possible SPM.

Claim 10 *If a node $t \in T_2$ has degree 1 in G', then it has to be matched in the SPM.*

PROOF: Suppose that t is not matched in the SPM. Let ts be the only edge incident to t in G', and let r be the root of the arborescence that s belongs to. By Claim 8, s is a leaf of this arborescence. If r is not matched along the branch of s, then s cannot be matched, and therefore st is blocking. If r is matched along the branch of s, then we exchange the edges along the branch and add ts and vr to the new matching, where vr is an edge of the arborescence from another branch; we also remove the original matching edge covering v. The new matching is at least as popular as the SPM, a contradiction. □

By the claim, if a node $t \in T_2$ has a single neighbor s in G', then we can fix ts and every second edge of the branch of s, and delete this branch and t. Suppose that this creates an arborescence with a single branch; then the original arborescence had two leaves, both in S (as we have already removed arborescences with only one leaf in S), and since the root cannot be matched on the branch of s, there is a unique way to match the whole arborescence. So in this case we can fix the matching on both branches

and remove the whole arborescence, maintaining the property that every arborescence has at least two branches.

After performing all of the above operations, the following hold.

- there are no parallel edges in G';
- every arborescence has at least two leaves in S;
- every leaf in S has at most one neighbor in T_2;
- every node in T_2 has degree at least 2;
- every node of each arborescence is matched in the SPM.

These properties can be satisfied only if every arborescence has exactly 2 leaves in S, every leaf in S has exactly one neighbor in T_2, and every node in T_2 has degree 2. This means that the graph contains a cycle if it is nonempty, and, in addition, every second edge in this cycle must be in the SPM. However, we can exchange along the cycle to get a new matching at least as popular as the SPM, which is a contradiction. We can conclude that the remaining graph is empty, which means that the only possible candidate for an SPM is F, i.e. the set of edges that we fixed. We can check in polynomial time if this is an SPM or not. □

Remark 11 *It is easy to see that we can apply the above algorithm with slight modifications to solve the following, slightly more general problem: nodes in S have strict preferences, and the preference lists of nodes in T can contain one tie, of arbitrary length, at the end. During the algorithm, we treat a node $t \in T$ as a node in T_P as long as the directed edge leaving t is not in the last tie. When the directed edge leaving t belongs to the last tie, we treat t as a node in T_I. In all other aspects, the algorithm remains the same.*

3 Conclusion

We proved that in case of strict preferences on one side and both strict preferences and indifference on the other side, the existence of a strongly popular matching can be decided in polynomial time. This is a clear indication that the strongly popular matching problem is significantly easier than the popular matching problem. It is difficult to complement this with hardness results; as mentioned in the introduction, the strongly popular matching problem is in the complexity class UP, for which no complete problems are known. Therefore, the more promising research direction is to consider polynomial-time solvability for slightly more general preference systems. In particular, the decision problem for strongly popular matchings is open in the following two cases:

- bipartite preference systems with strict preference and indifference allowed on both sides,
- bipartite preference systems with strict preferences on one side, and arbitrary preferences on the other side.

Our techniques do not seem to extend easily to these problems.

References

[1] D.J. Abraham, R.W. Irving, T. Kavitha, K. Mehlhorn, Popular matchings, *SIAM Journal on Computing* 37 (2007), 1030–1045.

[2] P. Biró, R. W. Irving, and D. F. Manlove, Popular matchings in the marriage and roommates problems, In *Proceedings of the 7th International Conference on Algorithms and Complexity*, volume 6078 of Lecture Notes in Computer Science (2010), 97–108. Full version: Technical Report TR-2009-306, University of Glasgow, Department of Computing Science.

[3] F. Brandl, T. Kavitha, Popular Matchings with Multiple Partners, *arXiv:* 1609.07531 (2016).

[4] Á. Cseh, C.-C. Huang, and T. Kavitha, Popular matchings with two-sided preferences and one-sided ties, In *Automata, Languages, and Programming*, M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, editors, volume 9134 of Lecture Notes in Computer Science (2015), 367–379. Full version: arXiv: 1603.07168

[5] Á. Cseh, Complexity and algorithms in matching problems under preferences, *Ph.D. Thesis*, TU Berlin (2016).

[6] Á. Cseh, T. Kavitha, Popular Edges and Dominant Matchings, In *Integer Programming and Combinatorial Optimization*, Volume 9682 of the series Lecture Notes in Computer Science (2016), 138–151.

[7] D. Gale, L.S. Shapley, College admissions and the stability of marriage, *American Mathematical Monthly*, 69 (1962), 9–15.

[8] P. Gärdenfors, Match making: assignments based on bilateral preferences, *Behavioural Science* 20 (1975), 166–173.

[9] C.-C. Huang, T. Kavitha, Popular Matchings in the Stable Marriage Problem, In *Automata, Languages and Programming*, Volume 6755 of the series Lecture Notes in Computer Science (2011), 666–677.

[10] C.-C. Huang, T. Kavitha, Near-Popular Matchings in the Roommates Problem, *SIAM J. Discrete Math.* 27 (2013), 43–62.

[11] R.W. Irving, An efficient algorithm for the "stable roommates" problem, *Journal of Algorithms* 6 (1985), 577–595.

[12] T. Kavitha, M. Nasre, Optimal popular matchings, *Discrete Applied Mathematics* 157 (2009), 3181–3186.

[13] E. McDermid, R.W. Irving, Popular matchings: Structure and algorithms, In *Proceedings of COCOON 2009*: the 15th Annual International Computing and Combinatorics Conference, volume 5609 of Lecture Notes in Computer Science (2009), 506–515.

[14] M. Nasre, A. Rawat, Popularity in the generalized Hospital Residents Setting, *arXiv:* 1609.07650 (2016).

[15] L. G. Valiant, Relative complexity of checking and evaluating, *Information Processing Letters* 5 (1976), 20–23.

On Applications of Weighted Linear Matroid Parity

Yusuke Kobayashi[1]

Division of Policy and Planning Sciences
University of Tsukuba
Tsukuba, Ibaraki 305-8573, Japan
kobayashi@sk.tsukuba.ac.jp

Yutaro Yamaguchi[2]

Department of
Information and Physical Sciences
Osaka University
Suita, Osaka 565-0871, Japan
yutaro_yamaguchi@ist.osaka-u.ac.jp

Abstract: A polynomial-time algorithm for the weighted linear matroid parity problem announced by Iwata (2013) in the second last symposium has been completed. We here demonstrate its powerfulness by showing several nontrivial, effective applications and possibilities.

Keywords: Linear matroid parity, Disjoint $\mathcal{S}$-paths, Feedback vertex sets

1 Introduction

The matroid parity problem was introduced by Lawler [19] as a unification of two fundamental generalizations of the bipartite matching problem: the non-bipartite matching problem and the matroid intersection problem. This problem cannot be solved in polynomial time in general [16, 20], but is known to be tractable as well as to admit a good characterization when the matroid in question is linearly represented due to Lovász [20, 21]; the problem is called the linear matroid parity problem.

While non-bipartite matching and matroid intersection can be solved in polynomial time also in reasonable weighted situations, the tractability of a weighted version of linear matroid parity had been open for a long while. Camerini, Galbiati, and Maffioli [4] first developed a randomized pseudopolynomial-time algorithm, which was later improved by Cheung, Lau, and Leung [5], but the running time bound is still pseudopolynomial. Recently, Iwata [14] and Pap [24] announced polynomial-time algorithms for weighted linear matroid parity, and the former work has been published as a full paper [15].

The linear matroid parity problem has a variety of applications in the sense that various combinatorial optimization problems can be solved efficiently through reductions to linear matroid parity: finding, e.g., a maximum number of disjoint $\mathcal{S}$-paths [21, 25], a minimum-cardinality feedback vertex set in a subcubic graph [26], a maximum-genus embedding of a graph [9], and a rooted-connected edge-orientation maximizing the number of vertices with even in-degree [8]. Such a reduction can be extended to weighted situations in a straightforward way in some cases (when there is a one-to-one correspondence between the feasible solutions before and after the reduction), but it sometimes fails. Such a trouble occurs when, for instance, a nontrivial transformation of solutions is required in addition to finding an optimal solution of a linear matroid parity instance. In this paper, we present how to overcome such difficulties by showing two interesting applications: the weighted versions of Mader's disjoint $\mathcal{S}$-paths problem and of the feedback vertex set problem in subcubic graphs.

The rest of this paper is organized as follows. Section 2 is devoted to formulating the linear matroid parity problem and its weighted versions. In Sections 3 and 4, we present how weighted linear matroid parity can solve the weighted versions of Mader's problem and of the feedback vertex set problem in subcubic graphs, respectively. Finally, we conclude this paper with further possible research directions in Section 5.

[1]Supported by JST, ERATO, Kawarabayashi Project, and by JSPS KAKENHI Grant Numbers 16K16010, 15H02966, and 16H03118.

[2]Supported by JSPS KAKENHI Grant Number 16H06931 and JST ACT-I Grant Number JPMJPR16UR.

2 Preliminaries

2.1 Linear matroid parity

Let $\mathbb{F}$ be a field, and Z a matrix over $\mathbb{F}$ whose row and column sets are U and V, respectively. We assume that the number $|V|$ of columns is even, and the column set V is partitioned into pairs of two distinct columns, called *lines*. Let L denote the set of lines. A column subset $X \subseteq V$ is called a *parity set* if X consists of lines, i.e., $|X \cap \ell| = 0$ or 2 for every line $\ell \in L$.

The linear independence of the column vectors of Z naturally defines a matroid on V (see, e.g., [23, 25] for the basic notions on matroids). We denote the linearly represented matroid by $\mathbf{M}(Z)$, whose independent set family, base family, and rank function are denoted by $\mathcal{I}(Z)$, $\mathcal{B}(Z)$, and r_Z, respectively. A base $B \in \mathcal{B}(Z)$ is called a *parity base* if B is a parity set. A set $X \subseteq V$ is said to be *spanning* if $r_Z(X) = r_Z(V)$.

The *linear matroid parity problem* is formulated as follows.

Linear Matroid Parity Problem

Input: A matrix $Z \in \mathbb{F}^{U \times V}$ over a field $\mathbb{F}$ with a line set L.

Goal: Find a maximum-cardinality independent parity set $I \in \mathcal{I}(Z)$.

Originated by Lovász [20], a variety of efficient algorithms for this problem have been developed; e.g., a deterministic augmenting-path algorithm by Gabow and Stallmann [10] and a randomized one by Cheung et al. [5].

2.2 Equivalent formulations of weighted linear matroid parity

For a weight $w \in \mathbb{R}^L$ defined on the line set L, we define the *weight* of a parity set $X \subseteq V$ as

$$w(X) := \sum_{\ell \in L\colon |X \cap \ell| = 2} w_\ell.$$

We first describe the following formulation of weighted linear matroid parity, which is adopted in [15].

Minimum-Weight Parity Base Problem

Input: A matrix $Z \in \mathbb{F}^{U \times V}$ over a field $\mathbb{F}$ with a line set L and a weight $w \in \mathbb{R}^L$.

Goal: Find a minimum-weight parity base $B \in \mathcal{B}(Z)$.

Theorem 1 (Iwata–Kobayashi [15, Theorem 11.1]) *The minimum-weight parity base problem can be solved with* $O(n^3 r)$ *arithmetic operations over* $\mathbb{F}$*, where* $n := |V|$ *and* $r := |U|$.

Another reasonable formulation is as follows, which is adopted in [24].

Maximum-Weight Independent Parity Set Problem

Input: A matrix $Z \in \mathbb{F}^{U \times V}$ over a field $\mathbb{F}$ with a line set L and a weight $w \in \mathbb{R}^L$.

Goal: Find a maximum-weight independent parity set $I \in \mathcal{I}(Z)$.

The next one is apparently different from the above two problems, but it turns out to be equivalent by considering the dual matroid $\mathbf{M}^*$ of $\mathbf{M}(Z)$, since a set $X \subseteq V$ is spanning in $\mathbf{M}(Z)$ if and only if $V \setminus X$ is independent in $\mathbf{M}^*$.

Minimum-Weight Spanning Parity Set Problem

Input: A matrix $Z \in \mathbb{F}^{U \times V}$ over a field $\mathbb{F}$ with a line set L and a weight $w \in \mathbb{R}^L$.

Goal: Find a minimum-weight spanning parity set $X \subseteq V$.

Remark 2 As shown in [20, Proposition 1.7], a minimum-*cardinality* spanning parity set can be constructed by adding an unspanned line repeatedly starting with any maximum-*cardinality* independent parity set. Hence, in the unweighted situation (i.e., when $w_\ell = 1$ for every line $\ell \in L$), a minimum spanning parity set is easily obtained by solving the same linear matroid parity instance (e.g., a minimum number of pinning-down points to make a planar structure rigid can be found in this way [21, Section 4]). This strategy, however, fails in the general weighted situation, and we have to consider the dual matroid explicitly, which may change the instance size (the number of rows) essentially.

3 Mader's Disjoint $\mathcal{S}$-paths

3.1 Background

Let $G = (V, E)$ be an undirected graph. For a prescribed vertex set $A \subseteq V$ with its partition $\mathcal{S}$ (i.e., $\mathcal{S}$ is a family of disjoint nonempty subsets of A whose union is A), an *A-path* is a path between distinct vertices in A that does not intersect A in between, and an *$\mathcal{S}$-path* is an A-path whose end vertices belong to distinct elements of $\mathcal{S}$. Each vertex in A is called a *terminal*.

Mader's disjoint $\mathcal{S}$-paths problem is to find a maximum number of vertex-disjoint $\mathcal{S}$-paths in a given undirected graph with a terminal set partitioned as $\mathcal{S}$. This problem also unifies two fundamental generalizations of bipartite matching: the non-bipartite matching problem and the disjoint s–t paths problem in undirected graphs.

Mader's problem was first mentioned by Gallai [11], and Mader [22] gave a good characterization by a min-max duality theorem. Lovász [20,21] proposed the first polynomial-time algorithm via a reduction to the matroid parity problem. In [21], instead of giving a linear representation, he showed how to handle the nontrivial double circuits that appear in this special case, and later Schrijver [25, Section 73.1a] provided an explicit linear representation, which leads to a direct reduction of Mader's problem to linear matroid parity.

Our goal in this section is to show the tractability of the weighted version of Mader's problem, which has been open for a long while similarly to weighted linear matroid parity. It does not immediately follow from the tractability of weighted linear matroid parity, but can be derived by a suitable transformation of the reduction for the unweighted case.

It should be remarked that Karzanov [18] presented a polynomial-time algorithm for a similar weighted problem in the edge-disjoint A-paths setting (which is a special case of Mader's setting), whose full proof was left to an unpublished paper [17]. Karzanov's problem can be solved by finding shortest k disjoint $\mathcal{S}$-paths (see Section 3.2) for all possible k, where the number of iterations is at most $|A|/2$ and can be reduced to $\mathrm{O}(\log |A|)$ by binary search.

3.2 Overview

In this section, we focus on the following weighted version of Mader's problem. For a family $\mathcal{P}$ of disjoint paths, we denote by $E(\mathcal{P})$ the set of edges traversed by some path in $\mathcal{P}$.

Shortest Disjoint $\mathcal{S}$-paths Problem

Input: An undirected graph $G = (V, E)$, a terminal set $A \subseteq V$ with its partition $\mathcal{S}$, a nonnegative edge length $c \in \mathbb{R}^E_{\geq 0}$, and a positive integer $k \in \mathbb{Z}_{>0}$.

Goal: Find a family $\mathcal{P}$ of k vertex-disjoint $\mathcal{S}$-paths in G with $c(\mathcal{P}) := \sum_{e \in E(\mathcal{P})} c_e$ minimum.

We shall reduce this problem to the minimum-weight parity base problem shown in Section 2.2. Our reduction results in an $\mathrm{O}(n) \times \mathrm{O}(n^2)$ matrix over the finite field $\mathbb{F}_p$ for some prime $p = \mathrm{O}(n)$ (see Remark 7), where $n := |V|$. Hence we can derive the following running time bound from Theorem 1 (we assume that each arithmetic operations over $\mathbb{F}_p$ can be performed in constant time by preparing the addition, multiplication, and inversion tables in advance).

Theorem 3 *The shortest disjoint $\mathcal{S}$-paths problem can be solved in $O(n^7)$ time, where $n := |V|$.*

Our reduction procedure is summarized as follows. We first construct an auxiliary graph G' from a given undirected graph G (see Section 3.4), which is the key point in this section. This step requires $O(n^2)$ time. Next, following Schrijver's linear representation, we make a matrix Z associated with the auxiliary graph G', and define a weight w from the edge length c in a natural way (see Section 3.5). This step takes $O(n^3)$ time. Finally, we show that the following two facts (see Claim 8), which complete the reduction within $O(n^3)$ time in total (and hence the computational time for weighted linear matroid parity is dominant):

- for any family $\mathcal{P}$ of k vertex-disjoint $\mathcal{S}$-paths in G, there exists a parity base B with $w(B) = c(\mathcal{P})$;
- for any parity base B, there exists a family $\mathcal{P}$ of k vertex-disjoint $\mathcal{S}$-paths in G with $c(\mathcal{P}) \leq w(B)$, which can be found easily, in $O(n)$ time.

In what follows, we sketch the detailed outline. For the complete proofs, we refer the readers to [27].

3.3 Associated matrix for Mader's $\mathcal{S}$-paths

In this section, we review Schrijver's reduction [25, Section 73.1a] of Mader's problem to linear matroid parity. For a given undirected graph $G = (V, E)$ with a terminal set $A \subseteq V$ partitioned as $\mathcal{S} = \{A_1, A_2, \ldots, A_t\}$, we construct an associated matrix $Z \in \mathbb{Q}^{2V \times 2E}$, where a 2×2 submatrix corresponds to each vertex $v \in V$ and each edge $e \in E$. Note that we use the field $\mathbb{Q}$ of rationals in this section for sake of simplicity, and it can be replaced by some finite field $\mathbb{F}_p$ (see Remark 7). We assume that every connected component of G has at least one $\mathcal{S}$-path.

Associate each edge $e = \{u, w\} \in E$ with a 2-dimensional linear subspace of $(\mathbb{Q}^2)^V$,

$$L_e := \{\, x \in (\mathbb{Q}^2)^V \mid x(u) + x(w) = \mathbf{0},\ x(v) = \mathbf{0}\ \ (v \in V \setminus \{u, w\})\,\}. \tag{1}$$

For each terminal $a \in A_i$ $(i = 1, 2, \ldots, t)$, define a 1-dimensional linear subspace

$$Q_a := \{\, x \in (\mathbb{Q}^2)^V \mid x(a) \in \left\langle \tbinom{1}{i} \right\rangle,\ x(v) = \mathbf{0}\ \ (v \in V \setminus \{a\})\,\}, \tag{2}$$

where $\langle y \rangle := \{\, py \mid p \in \mathbb{F}\,\}$ for a vector $y \in \mathbb{F}^r$ over a field $\mathbb{F}$.

Let $Q := \sum_{a \in A} Q_a$ and $\mathcal{E} := \{\, L_e/Q \mid e \in E\,\}$. Let us construct a matrix $Z \in \mathbb{Q}^{2V \times 2E}$ associated with $\mathcal{E}$ so that $\operatorname{rank} Z(F) = \dim(L_F/Q)$ for every $F \subseteq E$, where $Z(F) \in \mathbb{Q}^{2V \times 2F}$ denotes the submatrix of Z corresponding to F and $L_F := \sum_{e \in F} L_e$. This can be done by arranging an appropriate basis of $L_e/Q \in \mathcal{E}$ (which is regarded as taken from the original space $(\mathbb{Q}^2)^V$) for each edge $e \in E$ (see also Remark 6). Then, the independent parity sets for this Z are characterized as follows, from which we can derive a useful characterization of the parity bases (Lemma 5). Here we identify each edge set $F \subseteq E$ with the corresponding parity set for Z, which should be formally defined by the union of the column pairs corresponding to each edge in F.

Lemma 4 (Schrijver [25, (73.18)]) *An edge set $F \subseteq E$ is an independent parity set if and only if F forms a forest such that every connected component has at most two terminals in A, which belong to distinct elements of $\mathcal{S}$ if there are two.*

Lemma 5 *An edge set $F \subseteq E$ is a parity base if and only if F forms a spanning forest of G such that every connected component has exactly two terminals in A, which belong to distinct elements of $\mathcal{S}$.*

Remark 6 The above construction does not define a unique associated matrix $Z \in \mathbb{Q}^{2V \times 2E}$, and one is obtained as follows. We first compute the Kronecker product $B_G \otimes I_2 \in \mathbb{Q}^{2V \times 2E}$ of the incidence matrix $B_G \in \{-1, 0, 1\}^{V \times E} \subseteq \mathbb{Q}^{V \times E}$ of G (where each edge in G is assumed to be arbitrarily oriented) and the 2×2 identity matrix $I_2 \in \mathbb{Q}^{2 \times 2}$. Note that $B_G \otimes I_2$ is a matrix obtained by arranging a basis of L_e for each edge $e \in E$. We then obtain Z by adding to each column of $B_G \otimes I_2$ a multiple of a vector $x \in (\mathbb{Q}^2)^V$ with $\langle x \rangle = Q_a$ for each terminal $a \in A$ (e.g., x is defined by $x(a) := \binom{1}{i}$ and $x(v) := \mathbf{0}$ $(v \in V \setminus \{a\})$ when $a \in A_i$) so that all the entries in the first row of the corresponding submatrix $Z_a \in \mathbb{Q}^{2 \times 2E}$ are zero. This procedure takes $O(|V| \cdot |E|)$ time in total.

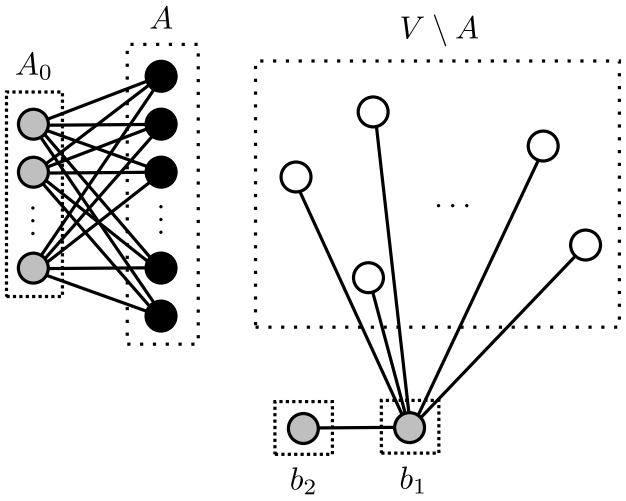


Figure 1: How to construct the auxiliary graph (the original edges are omitted).

Remark 7 The underlying field $\mathbb{Q}$ can be replaced by the finite field $\mathbb{F}_p$ for any prime $p > t$. This is because the essence of this representation is the linear independence of the two vectors $\binom{1}{i}$ and $\binom{1}{j}$ for every pair of i, j with $1 \leq i < j \leq t$. By the Bertrand–Chebyshev theorem, there exists a prime p such that $t < p < 2t$, and such a prime p can be found in $\mathrm{O}(t \log \log t)$ time by the sieve of Eratosthenes.

3.4 Construction of auxiliary graph

As the first step of our reduction, we construct an auxiliary undirected graph $G' = (V', E')$ with a terminal set $A' \subseteq V'$ partitioned as $\mathcal{S}'$ from a given undirected graph $G = (V, E)$ with a terminal set $A \subseteq V$ partitioned as $\mathcal{S}$. We assume that there exists a feasible solution, i.e., G has k vertex-disjoint $\mathcal{S}$-paths, and then we have $|A| \geq 2k$.

The construction is summarized as follows (see also Fig. 1). Add $(|A| - 2k)$ extra terminals so that each extra terminal is adjacent to all the original terminals in A, and let A_0 be the set of those extra terminals. Besides, add two other extra terminals b_1, b_2 so that b_1 and b_2 are adjacent and b_1 is adjacent to all the non-terminal vertices in $V \setminus A$. Finally, define $\mathcal{S}' := \mathcal{S} \cup \{A_0, \{b_1\}, \{b_2\}\}$.

Formally, for the vertex set, let a'_i $(i = 1, 2, \ldots, |A| - 2k)$ and b_j $(j = 1, 2)$ be distinct new vertices not in V, and define

$$
\begin{aligned}
A_0 &:= \{\, a'_i \mid i = 1, 2, \ldots, |A| - 2k \,\}, \\
V' &:= V \cup A_0 \cup \{b_1, b_2\}, \\
A' &:= A \cup A_0 \cup \{b_1, b_2\}, \\
\mathcal{S}' &:= \mathcal{S} \cup \{A_0, \{b_1\}, \{b_2\}\}.
\end{aligned}
$$

For the edge set, define

$$
\begin{aligned}
E_1 &:= \{\, e_{ia} = a'_i a \mid a'_i \in A_0,\ a \in A \,\}, \\
E_2 &:= \{\, e_v = b_1 v \mid v \in V \setminus A \,\}, \\
E' &:= E \cup E_1 \cup E_2 \cup \{e' = b_1 b_2\}.
\end{aligned}
$$

Note that, since $|A_0| \leq |A| = \mathrm{O}(n)$ and we may assume that G has no parallel edges, we have $|V'| = \mathrm{O}(n)$ and $|E'| = \mathrm{O}(|E| + n^2) = \mathrm{O}(n^2)$.

3.5 Completion of reduction

For the auxiliary graph $G' = (V', E')$ with a terminal set $A' \subseteq V'$ partitioned as $\mathcal{S}'$ obtained in Section 3.4, we construct an associated matrix $Z \in \mathbb{Q}^{2V' \times 2E'}$ defined in Section 3.3. Note that the construction requires $O(n^3)$ time, because $|V'| = O(n)$ and $|E'| = O(n^2)$. Define a weight $w \in \mathbb{R}^{E'}$ as follows: for each $e \in E'$,

$$w_e := \begin{cases} c_e & (e \in E), \\ 0 & (e \in E' \setminus E). \end{cases} \tag{3}$$

Note that $w(F) = w(F \cap E) = \sum_{e \in F \cap E} c_e$ for every $F \subseteq E'$, and $w(F_1) \leq w(F_2)$ for every $F_1 \subseteq F_2 \subseteq E'$ (recall that c is nonnegative), where a line set $F \subseteq E'$ and the corresponding parity set formed by F are identified.

Our reduction is completed by the following claim, which is shown based on Lemma 5. It implies that one can efficiently transform any minimum-weight parity base for Z into an optimal solution to the shortest disjoint $\mathcal{S}$-paths problem, and hence we conclude that Theorem 3 follows from Theorem 1. Recall that each edge set is identified with the corresponding parity set.

Claim 8 *The following relations hold between feasible solutions of the two problems.*

(i) *For any family $\mathcal{P}$ of k vertex-disjoint $\mathcal{S}$-paths in G, there exists a parity base $B_{\mathcal{P}} \subseteq E'$ with $B_{\mathcal{P}} \cap E = E(\mathcal{P})$ (hence, $w(B_{\mathcal{P}}) = c(\mathcal{P})$).*

(ii) *For any parity base $B \subseteq E'$, there exists a family $\mathcal{P}_B$ of k vertex-disjoint $\mathcal{S}$-paths in G with $E(\mathcal{P}_B) \subseteq B \cap E$ (hence, $c(\mathcal{P}_B) \leq w(B)$), which can be found in $O(n)$ time.*

4 Feedback Vertex Sets in Subcubic Graphs

4.1 Overview

Let $G = (V, E)$ be an undirected graph. A vertex set $X \subseteq V$ is called a *feedback vertex set* if its removal results in a forest, i.e., every cycle in G intersects at least one vertex in X. The problem of determining the minimum cardinality of a feedback vertex set in a given graph is NP-complete, even if we are restricted ourselves to the planar graphs of maximum degree four [12].

In contrast, Ueno, Kajitani, and Gotoh [26] showed that, if the input graph is of maximum degree three (we say such a graph to be *subcubic*), a minimum-cardinality feedback vertex set can be found in polynomial time via linear matroid parity. The reduction was given in a two-phased form as follows (see Section 4.3 for the details): they first showed that a maximum-cardinality nonseparating independent set in a subcubic graph can be found by solving the linear matroid parity problem, and then constructed a minimum-cardinality feedback vertex set by adding some vertices to it in a suitable manner.

Our target in this section is the following weighted version of the feedback vertex set problem.

Minimum-Weight Feedback Vertex Set Problem

Input: An undirected graph $G = (V, E)$ and a nonnegative vertex weight $c \in \mathbb{R}^V_{\geq 0}$.

Goal: Find a minimum-weight feedback vertex set $X \subseteq V$.

For the general case, a polynomial-time 2-approximation algorithm [2] is best known. When the treewidth of an input graph is bounded, the problem can be solved in polynomial time by a dynamic programming strategy [3]. For more details, we refer the reader to [7].

We shall reduce this problem on subcubic graphs to the maximum-weight independent parity set problem. Our reduction is via another problem that was formulated as linear matroid parity by Lovász [21, Section 2], called the 3-forest problem (see Section 4.2). As a result, Theorem 1 leads to the following running time bound.

Theorem 9 *The minimum-weight feedback vertex set problem in subcubic graphs can be solved in $O(n^4)$ time, where $n := |V|$.*

4.2 Reduction to 3-forest problem

Let $H = (W, F)$ be a 3-uniform hypergraph, i.e., F is a family of 3-element subsets of W. We denote by $B(H) = (W, F; A)$ the bipartite graph representing H, i.e., $A = \{ (w, f) \mid w \in f \in F \}$. A hyperedge set $F' \subseteq F$ is called a 3-*forest* if the subgraph of $B(H)$ induced by $F' \cup \bigcup_{f \in F'} f$ is a forest in the usual sense. Lovász [21, Section 2] showed that a maximum 3-forest in a given 3-uniform hypergraph can be found via linear matroid parity.

Theorem 10 (Lovász) *Let $H = (W, F)$ be a 3-uniform hypergraph. There exists a linear matroid with the line set F such that a subset $F' \subseteq F$ forms an independent parity set if and only if F' is a 3-forest.*

Remark 11 A linear representation of this matroid is obtained as follows. Associate each hyperedge $f = \{u, v, w\} \in F$ with a 2-dimensional linear subspace of $\mathbb{F}_2^W$,

$$L_f := \{ x \in \mathbb{F}_2^W \mid x(u) + x(v) + x(w) = \mathbf{0},\ x(t) = \mathbf{0} \ \ (t \in W \setminus \{u, v, w\}) \}.$$

Let $Z \in \mathbb{F}_2^{W \times 2F}$ be a matrix obtained by arranging a basis of L_f for each $f \in F$, which consists of vectors in $\mathbb{F}_2^W$ with exactly two nonzero entries at the three candidates corresponding to f.

In what follows, we show that the minimum-weight feedback vertex set problem in subcubic graphs reduces to the following weighted version of the 3-forest problem, which is a special case of the maximum-weight independent parity set problem shown in Section 2.2 (as an immediate consequence of Theorem 10).

Maximum-Weight 3-Forest Problem

Input: A 3-uniform hypergraph $H = (W, F)$ and a weight $w \in \mathbb{R}^F$.

Goal: Find a maximum-weight 3-forest $F' \subseteq F$.

Let $G = (V, E)$ and $c \in \mathbb{R}^V_{\geq 0}$ be the input of the minimum-weight feedback vertex set problem, and suppose that G is subcubic. We first subdivide each edge $e = \{u, v\} \in E$, i.e., remove the edge e and add a new vertex w_e and two new edges $\{w_e, u\}$ and $\{w_e, v\}$. Let W be the set of new vertices. Then the resulting graph is a bipartite graph with the color sets W and V, say $B = (W, V; A)$.

Since G is subcubic, each original vertex $v \in V$ has at most three neighbors in W in B. We may assume that v has exactly three neighbors by the following procedure: if the number of the neighbors of v is $k < 3$, then we add $3 - k$ extra vertices to W so that they are adjacent only to v. Let $H = (W, V)$ be the 3-uniform hypergraph represented by B.

Our reduction is completed by the following claim, whose proof is just an exercise. With the aid of the linear representation given in Remark 11 (and the equivalence of the three formulations described in Section 2.2), we can conclude that Theorem 9 follows from Theorem 1. Note that $|W| \leq 3|V| = \mathrm{O}(n)$.

Claim 12 *A hyperedge set $X \subseteq V$ is a 3-forest in H if and only if $V \setminus X$ is a feedback vertex set in G.*

4.3 Relation to nonseparating independent set problem

The solution to the unweighted version given by Ueno et al. [26] was via the nonseparating independent set problem. In a graph $G = (V, E)$, a vertex set $X \subseteq V$ is called an *independent set* if X induces no edge in E (i.e., no edge in E connects two distinct vertices in X). In addition, X is said to be *nonseparating* if the removal of any subset of X does not increase the number of connected components.

Ueno et al. [26] showed a reduction of finding a maximum-cardinality nonseparating independent set in a subcubic graph to the linear matroid parity problem, and that a minimum-cardinality feedback vertex set in a subcubic graph can be easily constructed from an optimal solution of the former problem.

Theorem 13 (Ueno–Kajitani–Gotoh [26, Theorems 1 and 2]) *Let $G = (V, E)$ be a subcubic graph. There exists a linear matroid $\mathbf{M}$ with the line set V such that a subset $X \subseteq V$ forms a independent parity set in $\mathbf{M}$ if and only if X is a nonseparating independent set in G.*

Remark 14 Such a linear matroid is obtained as follows. Let $\mathbf{M}_G$ be the cycle matroid on E, i.e., a subset $F \subseteq E$ is independent if and only if F forms a forest. Let $\mathbf{M}_G^*$ be the dual matroid of $\mathbf{M}_G$, i.e., a subset $F \subseteq E$ is independent if and only if $E \setminus F$ forms a maximal forest in G. For each vertex $v \in V$, the set $\delta_G(v)$ of edges incident to v is of rank at most two in $\mathbf{M}_G^*$, because the removal of those three edges increases the number of connected components. Hence, regardless of its linear representation, the dimension of the subspace U_v spanned by $\delta_G(v)$ is at most two. Take a basis of U_v and associate v with the two vectors in it (if it consists of at most one vector, then add the zero vector). The resulting linear matroid with the line set V is a desired one.

Theorem 15 (Ueno–Kajitani–Gotoh [26, Section 3.2]) *Let $G = (V, E)$ be a subcubic graph, and $X \subseteq V$ a maximum-cardinality nonseparating independent set in G. Then every 2-connected component of $G - X$ has at most one cycle, and one can construct a minimum-cardinality feedback vertex set in G by adding one vertex in each of such remaining cycles to X.*

This strategy itself cannot be extended to the weighted situation (cf. Remark 2), but they also observed the following important fact.

Theorem 16 (Ueno–Kajitani–Gotoh [26, Theorem 3]) *Let $G = (V, E)$ be a subcubic graph. A subset $X \subseteq V$ forms a spanning parity set in the linear matroid in Remark* 14 *if and only if X is a feedback vertex set in G.*

This leads to a natural extension to the weighted situation with the aid of one of the equivalent formulations of weighted linear matroid parity given in Section 2.2, the minimum-weight spanning parity set problem. Moreover, it can be confirmed that the two linear matroids obtained in Sections 4.2 and 4.3 (i.e., via a reduction to the 3-forest problem and Remark 11, and via a reduction to the nonseparating independent set problem and Remark 14) are indeed the dual one of each other.

5 Concluding Remarks

In this paper, we have presented successful applications of weighted linear matroid parity. As a main result, we have given a positive answer to a longstanding open problem: whether the weighted version of Mader's disjoint $\mathcal{S}$-paths problem is tractable or not. This result brings the following natural questions: can we obtain (a) a min-max duality theorem (extending Mader's theorem [22] in the unweighted case), and (b) an integral polyhedral description of disjoint $\mathcal{S}$-paths (like Edmonds' matching polytope [6])? It might be natural to conjecture that the answers are both YES, but our reduction itself cannot help to prove those right away because neither min-max duality theorem nor integral polyhedral description is known for weighted linear matroid parity.

We also suspect that there are other kinds of interesting applications that do not occur as weighted versions of problems originally solved via unweighted linear matroid parity. A possible candidate is the *simplex matching problem* [1]. An inattentive unweighted version of this problem includes the 3-dimensional matching problem as well as non-bipartite matching, and hence is NP-hard [13]. With a reasonable restriction on the weight, however, the problem becomes tractable. If one could handle such a situation by graph and linear matroid tricks, we would interpret the tractability (which may be somewhat surprising in appearance) from a different point of view.

References

[1] E. Anshelevich, A. Karagiozova: Terminal backup, 3D matching, and covering cubic graphs. *SIAM Journal on Computing*, **40** (2011), pp. 678–708.

[2] V. Bafna, P. Berman, T. Fujito: A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM Journal on Discrete Mathematics*, **12** (1999), pp. 289–297.

[3] H. L. Bodlaender: Dynamic programming on graphs with bounded treewidth. *Proceedings of the 15th International Colloquium on Automata, Languages, and Programming* (*ICALP 1988*), pp. 105–118, 1988.

[4] P. M. Camerini, G. Galbiati, F. Maffioli: Random pseudo-polynomial algorithms for exact matroid problems. *Journal of Algorithms*, **13** (1992), pp. 258–273.

[5] H. Y. Cheung, L. C. Lau, K. M. Leung: Algebraic algorithms for linear matroid parity problems. *ACM Transactions on Algorithms*, **10** (2014), No. 10.

[6] J. Edmonds: Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards, Sec. B*, **69** (1965), pp. 125–130.

[7] P. Festa, P. M. Pardalos, M. G. C. Resende: Feedback set problems. In *Handbook of Combinatorial Optimization*, Supplement Volume A, Springer, 1999, pp. 209–258.

[8] A. Frank, T. Jordán, Z. Szigeti: An orientation theorem with parity conditions. *Discrete Applied Mathematics*, **115** (2001), pp. 37–47.

[9] M. L. Furst, J. L. Gross, L. A. McGeoch: Finding a maximum-genus graph imbedding. *Journal of the ACM*, **35** (1988), pp. 523–534.

[10] H. N. Gabow, M. Stallmann: An augmenting path algorithm for linear matroid parity. *Combinatorica*, **6** (1986), pp. 123–150.

[11] T. Gallai: Maximum-minimum Sätze und verallgemeinerte Faktoren von Graphen. *Acta Mathematica Academiae Scientiarum Hungaricae*, **12** (1961), pp. 131–173.

[12] M. R. Garey, D. S. Johnson: The rectilinear steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, **32** (1977), pp. 826–834.

[13] M. R. Garey, D. S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, 1979.

[14] S. Iwata: A weighted linear matroid parity algorithm. *Proceedings of the 8th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications*, pp. 251–259, 2013.

[15] S. Iwata, Y. Kobayashi: A weighted linear matroid parity algorithm. *Mathematical Engineering Technical Reports*, METR 2017-01, University of Tokyo, 2017.

[16] P. Jensen, B. Korte: Complexity of matroid property algorithms. *SIAM Journal on Computing*, **11** (1982), pp. 184–190.

[17] A. V. Karzanov: Edge-disjoint T-paths of minimum total cost. *Technical Report*, STAN-CS-92-1465, Department of Computer Science, Stanford University, 1993.

[18] A. V. Karzanov: Multiflows and disjoint paths of minimum total cost. *Mathematical Programming*, **78** (1997), pp. 219–242.

[19] E. L. Lawler: *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976.

[20] L. Lovász: The matroid matching problem. *Colloquia Mathematica Societatis János Bolyai*, **25** (1978), pp. 495–517.

[21] L. Lovász: Matroid matching and some applications. *Journal of Combinatorial Theory, Ser. B*, **28** (1980), pp. 208–236.

[22] W. Mader: Über die Maximalzahl krezungsfreier H-Wege. *Archiv der Mathematik*, **31** (1978), pp. 387–402.

[23] J. Oxley: *Matroid Theory*, 2nd ed., Oxford University Press, 2011.

[24] G. Pap: Weighted linear matroid matching. *Proceedings of the 8th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications*, pp. 411–413, 2013.

[25] A. Schrijver: *Combinatorial Optimization — Polyhedra and Efficiency*, Springer-Verlag, 2003.

[26] S. Ueno, Y. Kajitani, S. Gotoh: On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. *Discrete Mathematics*, **72** (1988), pp. 355–360.

[27] Y. Yamaguchi: Shortest disjoint $\mathcal{S}$-paths via weighted linear matroid parity. *Proceedings of the 27th International Symposium on Algorithms and Computation* (*ISAAC 2016*), No. 63, 2016.

A note on a conjecture about shattering-extremal set systems

Christopher Kusch[1]

Department of Mathematics and Computer Science, Institute of Mathematics
Freie Universität Berlin
Arnimalle 3, 14195 Berlin, Germany
chrishk@zedat.fu-berlin.de

Tamás Mészáros[2]

Department of Mathematics and Computer Science, Institute of Mathematics
Freie Universität Berlin
Arnimalle 3, 14195 Berlin, Germany
tamas.meszaros@fu-berlin.de

Abstract: We say that a set system $\mathcal{F} \subseteq 2^{[n]}$ *shatters* a given set $S \subseteq [n]$ if

$$2^S = \{F \cap S : F \in \mathcal{F}\}.$$

The Sauer-Shelah lemma states that in general, a set system $\mathcal{F}$ shatters at least $|\mathcal{F}|$ sets. Here we concentrate on the case of equality. A set system is called *shattering-extremal* if it shatters exactly $|\mathcal{F}|$ sets. In this note we investigate the problem of adding sets to shattering-extremal families so that the resulting family is still shattering-extremal.

Keywords: shattering, Sauer-Shelah lemma, shattering-extremal set systems

1 Introduction

Let $n \in \mathbb{N}$ and set $[n] = \{1, ..., n\}$. If $X \subseteq [n]$ and $I \subseteq [n] \setminus X$, we write 2^X to denote the power set of X, $I + 2^X$ for the family $\{I \cup A : A \subseteq X\}$ and $\binom{X}{k}$ for the collection of subsets of X of size k. A set system $\mathcal{F} \subseteq 2^{[n]}$ is a *down-set* (*up-set*) if $G \subseteq F$ and $F \in \mathcal{F}$ ($G \in \mathcal{F}$) implies $G \in \mathcal{F}$ ($F \in \mathcal{F}$). A set system *shatters* a given set $S \subseteq [n]$ if

$$2^S = \{F \cap S : F \in \mathcal{F}\}.$$

The family of subsets of $[n]$ shattered by $\mathcal{F}$ is denoted by $\mathrm{Sh}(\mathcal{F})$.

Proposition 1 *$|Sh(\mathcal{F})| \geq |\mathcal{F}|$ for every set system $\mathcal{F} \subseteq 2^{[n]}$.*

This statement was proved by several authors independently (e.g. [17],[18],[19]), and is often referred to as the Sauer-Shelah lemma. For a proof see e.g. [3]. Here we are interested in the case of equality. A set systems $\mathcal{F} \subseteq 2^{[n]}$ is *shattering-extremal*, or *s-extremal* for short, if it shatters exactly $|\mathcal{F}|$ sets, i.e. $|\mathcal{F}| = |\mathrm{Sh}(\mathcal{F})|$. For example, if $\mathcal{F}$ is a down-set then $\mathcal{F}$ is s-extremal, simply because in this case $\mathrm{Sh}(\mathcal{F}) = \mathcal{F}$. Many interesting results have been obtained in connection with these combinatorial objects, among others by Bollobás, Leader and Radcliffe in [4], by Bollobás and Radcliffe in [5], by Frankl in [7] and recently Kozma and Moran in [10] provided further interesting examples of s-extremal set systems. Anstee, Rónyai and Sali in [3] related shattering to standard monomials of vanishing ideals, and based on this, Mészáros and Rónyai in [15] developed algebraic methods for the investigation of s-extremal families, which we will briefly recall later.

[1]Research is supported by the Berlin Mathematical School Phase II program
[2]Research is supported by the DRS POINT Postdoc Fellow program

To broaden the picture, we now mention some well known related results. The *Vapnik-Chervonenkis dimension* of $\mathcal{F}$, denoted by $\dim_{VC}(\mathcal{F})$, is the size of the largest set shattered by $\mathcal{F}$. An easy corollary of the Sauer-Shelah lemma is the following result, known as the Sauer-inequality, which has found applications in a variety of contexts.

Proposition 2 ([17],[18],[19]) *Let $0 \leq k \leq n$ and $\mathcal{F} \subseteq 2^{[n]}$. If $\mathcal{F}$ shatters no set of size k, i.e. $dim_{VC}(\mathcal{F}) \leq k-1$, then*

$$|\mathcal{F}| \leq \sum_{i=1}^{k-1} \binom{n}{i}. \tag{1}$$

Families satisfying (1) with equality are called *maximum classes*, and serve as important examples in the theory of machine learning. They have several nice properties, among others they are s-extremal. In the case of uniform families the above bound can be strengthened.

Proposition 3 ([8]) *Let $0 \leq k \leq l \leq n$ and $\mathcal{F} \subseteq \binom{[n]}{l}$. If $\mathcal{F}$ shatters no set of size k, i.e. $dim_{VC}(\mathcal{F}) \leq k-1$, then*

$$|\mathcal{F}| \leq \binom{n}{k-1}.$$

A set family $\mathcal{S} \subseteq 2^{[n]}$ is called a *Sperner family*, or an *antichain*, if none of its sets is contained in another. We define the up-set generated by $\mathcal{S}$ as

$$\mathrm{Up}(\mathcal{S}) = \{F \subseteq [n] \ : \ \exists S \in \mathcal{S} \text{ such that } S \subseteq F\}.$$

In connection with Proposition 3 it is an interesting open problem whether the above bound holds for Sperner families in general not merely uniform ones.

Now let us return to the study of s-extremal families. The main goal here is to find good characterizations of them. A positive answer to the following conjecture, formulated in [12], would be a possible way for this.

Conjecture 4 *For every s-extremal set system $\mathcal{F} \subsetneq 2^{[n]}$ there exists $F \notin \mathcal{F}$ such that $\mathcal{F} \cup \{F\}$ is again s-extremal.*

As by Theorem 2 in [5] $\mathcal{F}$ is s-extremal if and only if $2^{[n]} \setminus \mathcal{F}$ is so, the above conjecture has an equivalent form, namely that for every non-empty s-extremal set system $\mathcal{F} \subseteq 2^{[n]}$ there exists $F \in \mathcal{F}$ such that $\mathcal{F} \setminus \{F\}$ is again s-extremal. It will be always clear from the context which form of the conjecture we consider. This latter form was formulated by Litman and Moran independently, and called the corner peeling conjecture. For maximum classes essentially the same was conjectured by Kuzmin an Warmuth in [11] and proven by Rubinstein and Rubinstein in [16]. There are several other cases when the conjecture is known to be true. First of all it is trivially true for down-sets, as there you can always add any minimal element not belonging to it. Mészáros and Rónyai in [12] and [13], using a graph theoretic approach, proved the conjecture for s-extremal families of VC-dimension at most 2. According to personal communication, the same result was independently proven by Litman and Moran. Some examples of Anstee in [2] and of Füredi and Quinn in [9] also turned out to be s-extremal and they also satisfy the conjecture. According to Moran and Warmuth, [14] the conjecture, if true, would imply unlabeled compression schemes for s-extremal classes, which so far were known to exist for maximum classes.

In this note we prove Conjecture 4 for a further class of s-extremal systems which we introduce in the next section using the algebraic approach from [15]. We also relate this approach to a generalization of the Sauer inequality.

2 Preliminaries

Let $\mathbb{F}$ be an arbitrary field and let $\mathbb{F}[x_1, ..., x_n] = \mathbb{F}[\mathbf{x}]$ be the polynomial ring over $\mathbb{F}$ with variables $x_1, ..., x_n$. Given some set $F \subseteq [n]$, let $v_F \in \{0,1\}^n$ be its *characteristic vector*, i.e. the i-th coordinate of v_F is 1 if $i \in F$ and 0 otherwise. Therefore we can identify a set system $\mathcal{F} \subseteq 2^{[n]}$ with the vector system

$$\mathcal{V}(\mathcal{F}) = \{v_F \ : \ F \in \mathcal{F}\} \subseteq \{0,1\}^n \subseteq \mathbb{F}^n.$$

One can then associate to $\mathcal{F}$ a polynomial ideal $I(\mathcal{V}(\mathcal{F})) \trianglelefteq \mathbb{F}[\mathbf{x}]$, where

$$I(\mathcal{F}) = I(\mathcal{V}(\mathcal{F})) = \{f \in \mathbb{F}[\mathbf{x}] \ : \ f(v_F) = 0 \ \forall \ F \in \mathcal{F}\}.$$

In words, $I(\mathcal{F})$ is the vanishing ideal of the set of characteristic vectors of the elements of $\mathcal{F}$. Note that we always have $\{x_i^2 - x_i \ : \ i \in [n]\} \subseteq I(\mathcal{F})$. For more details about vanishing ideals of finite point sets see e.g. [15].

If one works with polynomial ideals, it is useful to have a nice ideal basis. Such nice bases are given by the so-called *Gröbner bases*, which we will now briefly define. For more details the interested reader may consult e.g. [1]. A total order $\prec$ on the monomials in $\mathbb{F}[\mathbf{x}]$ is a *term order*, if 1 is the minimal element of $\prec$, and $\prec$ is compatible with multiplication with monomials. One well-known and important term order is the *lexicographic (lex) order*. Here one has $x_1^{w_1}...x_n^{w_n} \prec_{\text{lex}} x_1^{u_1}...x_n^{u_n}$ if and only if for the smallest index k with $w_k \neq u_k$ one has $w_k < u_k$. One can build a lex order based on other orderings of the variables as well, so altogether we have $n!$ different lex orders. Given some term order $\prec$ and $f \in \mathbb{F}[\mathbf{x}]$, the *leading monomial* $\text{Lm}(f)$ of f, is the largest monomial (with respect to $\prec$) appearing with non-zero coefficient in the canonical form of f.

Now let $I \trianglelefteq \mathbb{F}[\mathbf{x}]$ be an ideal and $\prec$ a term order. A finite subset $\mathbb{G} \subseteq I$ is called a *Gröbner basis of* I with respect to $\prec$ if for every $f \in I$ there exists a $g \in \mathbb{G}$ such that $\text{Lm}(g)$ divides $\text{Lm}(f)$. If for every $g \in \mathbb{G}$ we have that $\text{Lm}(g)$ does not divide any monomial appearing with non-zero coefficient in any other polynomial in $\mathbb{G}$, then $\mathbb{G}$ is called a reduced Gröbner basis. Gröbner bases have a lot of nice properties, among others we know that every non-zero ideal $I \trianglelefteq \mathbb{F}[\mathbf{x}]$ has a unique reduced Gröbner basis for every term order, and if $\mathbb{G}$ is a Gröbner basis of I for some term order, then $\mathbb{G}$ generates I as an ideal as well, i.e. $I = \langle \mathbb{G} \rangle$.

For a subset $H \subseteq [n]$, set $\mathbf{x}_H = \prod_{i \in H} x_i$. Given a pair of sets $H \subseteq S \subseteq [n]$ we then define the polynomial

$$f_{S,H}(\mathbf{x}) = \mathbf{x}_H \cdot \prod_{i \in S \setminus H} (x_i - 1).$$

A nice property of these polynomials is that $\text{Lm}(f_{S,H}) = x_S$ for every term order and for a set $F \subseteq [n]$ we have $f_{S,H}(v_F) \neq 0$ if and only if $F \cap S = H$.

Now we are in a position to state the connection between s-extremal families and the theory of Gröbner bases.

Theorem 5 ([15]) *$\mathcal{F} \subseteq 2^{[n]}$ is s-extremal if and only if there are polynomials of the form $f_{S,H}$, which together with $\{x_i^2 - x_i \ : \ i \in [n]\}$ form a (reduced) Gröbner basis of $I(\mathcal{F})$ for all term orders.*

From the proof of Theorem 5 from [15] one can deduce some additional properties, which we will summarize in the following remark.

Remark 6
a) If there is a suitable Gröbner basis for one particular term order, then $\mathcal{F}$ is already s-extremal.
b) Suppose $\mathcal{F}$ is s-extremal, and let $\mathcal{S}$ be the collection of all sets S from the Gröbner basis. If we require the Grbner basis to be reduced, we have that $\mathcal{S}$ is a Sperner system. More precisely $\mathcal{S}$ is the collection of all minimal sets that are not shattered by $\mathcal{F}$. In particular $\mathcal{S}$ is a Sperner family and $Sh(\mathcal{F}) = 2^{[n]} \setminus Up(\mathcal{S})$. Moreover, given $S \in \mathcal{S}$ the corresponding H is its unique subset for which there does not exist an $F \in \mathcal{F}$ with $F \cap S = H$. In particular this means that the reduced Gröbner basis in Theorem 5 is unique.

In accordance with these results we introduce some further notation. Suppose we are given a Sperner family $\mathcal{S} \subseteq 2^{[n]}$ and a function $h : \mathcal{S} \to 2^{[n]}$ such that $h(S) \subseteq S$ for every $S \in \mathcal{S}$. For $H \subseteq S \subseteq [n]$ define

$$\mathcal{P}_S = S + 2^{[n]\setminus S} \text{ and } \mathcal{Q}_{S,H} = H + 2^{[n]\setminus S}.$$

Note that $\mathcal{P}_S$ and $\mathcal{Q}_{S,h(S)}$ are hypercubes of the same dimension, in particular $|\mathcal{P}_S| = |\mathcal{Q}_{S,h(S)}|$. Further set

$$\mathcal{H}(\mathcal{S}) = 2^{[n]} \setminus \mathrm{Up}(\mathcal{S}) = 2^{[n]} \setminus \bigcup_{S\in\mathcal{S}} \mathcal{P}_S,$$

$$\mathcal{F}(\mathcal{S},h) = 2^{[n]} \setminus \bigcup_{S\in\mathcal{S}} \mathcal{Q}_{S,h(S)} \text{ and}$$

$$\mathbb{G}(\mathcal{S},h) = \{f_{S,h(S)} \ : \ S \in \mathcal{S}\} \cup \{x_i^2 - x_i \ : \ i \in [n]\}.$$

3 Main results

Now we are able to state our main results. Let $\mathcal{S} \subseteq 2^{[n]}$ be a Sperner family and h a function as above.

Proposition 7 *$\mathbb{G} = \mathbb{G}(\mathcal{S},h)$ is a Gröbner basis (of $\langle\mathbb{G}\rangle$) for some term order $\prec$ if and only if*

$$|\mathcal{H}(\mathcal{S})| = |\mathcal{F}(\mathcal{S},h)|.$$

Proof: Suppose first that $\mathbb{G}$ is a Gröbner basis for some term order $\prec$. Then standard arguments from algebra (for details see e.g. the proof of Theorem 9 in [12]) show that $\langle\mathbb{G}\rangle$ is a radical ideal and $\langle\mathbb{G}\rangle = I(\mathcal{F})$ where $\mathcal{F}$ (more precisely $\mathcal{V}(\mathcal{F})$) is the set of common roots of the polynomials in $\mathbb{G}$. By the earlier mentioned properties of the $f_{S,h(S)}$ polynomials we have that

$$\mathcal{F} = \bigcap_{S\in\mathcal{S}} \{F \ : \ v_F \text{ is a root of } f_{S,h(S)}\} = \bigcap_{S\in\mathcal{S}} \{F \ : \ F \cap S \neq h(S)\}$$

$$= \{F \ : \ F \cap S \neq h(S) \ \forall S \in \mathcal{S}\} = 2^{[n]} \setminus \bigcup_{S\in\mathcal{S}} \mathcal{Q}_{S,h(S)} = \mathcal{F}(\mathcal{S},h).$$

Thus $\langle\mathbb{G}\rangle = I(\mathcal{F}(\mathcal{S},h))$ and so, by Theorem 5, $\mathcal{F}(\mathcal{S},h)$ is s-extremal, i.e. $|\mathcal{F}(\mathcal{S},h)| = |\mathrm{Sh}(\mathcal{F}(\mathcal{S},h))|$. However by Remark 6 in this case we have that $\mathrm{Sh}(\mathcal{F}(\mathcal{S},h)) = \mathcal{H}(\mathcal{S})$, and so $|\mathcal{F}(\mathcal{S},h)| = |\mathcal{H}(\mathcal{S})|$.

Now suppose $|\mathcal{H}(\mathcal{S})| = |\mathcal{F}(\mathcal{S},h)|$. In terms of Theorem 5 it is enough to show that $\mathcal{F}(\mathcal{S},h)$ is s-extremal. Note that by definition, for every $S \in \mathcal{S}$ there does not exist $F \in \mathcal{F}(\mathcal{S},h)$ such that $F \cap S = h(S)$ and so $S \notin \mathrm{Sh}(\mathcal{F}(\mathcal{S},h))$. In particular no superset of S is shattered by $\mathcal{F}(\mathcal{S},h)$. Therefore $\mathrm{Sh}(\mathcal{F}(\mathcal{S},h)) \subseteq 2^{[n]} \setminus \mathrm{Up}(\mathcal{S}) = \mathcal{H}(\mathcal{S})$ and hence $|\mathrm{Sh}(\mathcal{F}(\mathcal{S},h))| \leq |\mathcal{H}(\mathcal{S})| = |\mathcal{F}(\mathcal{S},h)|$. However the opposite inequality holds by the Sauer-Shelah lemma for every set system and thus $\mathcal{F}(\mathcal{S},h)$ is necessary s-extremal. □

Using Remark 6, the following is an immediate consequence.

Corollary 8 *$\mathcal{F} = \mathcal{F}(\mathcal{S},h)$ is s-extremal with $Sh(\mathcal{F}) = \mathcal{H}(\mathcal{S})$ if and only if*

$$|\mathcal{H}(\mathcal{S})| = |\mathcal{F}(\mathcal{S},h)|. \tag{2}$$

Here one should note that by Remark 6 every s-extremal family $\mathcal{F} \subseteq 2^{[n]}$ is of the form $\mathcal{F}(\mathcal{S},h)$ for some well defined Sperner system $\mathcal{S}$ and function h, so this corollary really might be a good first step towards a nice characterization of s-extremal families.

To try to justify this approach a bit further we remark that it has a connection to the following generalization of the Sauer inequality which was implicitly proved in the proof of Proposition 7. To emphasize it below we shortly repeat the argument. We remark that our attention to this way of generalizing the Sauer inequality was raised by Chornomaz, [6].

Proposition 9 *Let $\mathcal{S} \subseteq 2^{[n]}$ be a Sperner family and $\mathcal{F} \subseteq 2^{[n]}$ a set system that shatters no element of $\mathcal{S}$. Then*

$$|\mathcal{F}| \leq |\mathcal{H}(\mathcal{S})|.$$

PROOF: For the proof just note that if $\mathcal{F}$ shatters no element of $\mathcal{S}$, then it shatters no set from $\mathrm{Up}(\mathcal{S})$ either, and so $\mathrm{Sh}(\mathcal{F}) \subseteq 2^{[n]} \setminus \mathrm{Up}(\mathcal{S})$. Accordingly

$$|\mathcal{F}| \leq |\mathrm{Sh}(\mathcal{F})| \leq |2^{[n]} \setminus \mathrm{Up}(\mathcal{S})| = |\mathcal{H}(\mathcal{S})|$$

as wanted. □

For a Sperner family $\mathcal{S} \subseteq 2^{[n]}$ let us define a family $\mathcal{F} \subseteq 2^{[n]}$ shattering no element of $\mathcal{S}$ and satisfying $|\mathcal{F}| = |\mathcal{H}(\mathcal{S})|$ to be $\mathcal{S}$-*extremal*. Note that the original Sauer inequality can be recovered by setting $\mathcal{S} = \binom{[n]}{k}$, and $\binom{[n]}{k}$-extremal families are just the maximum classes. An interesting property here is that if we let $\mathcal{S}$ to vary, then we end up with s-extremality.

Proposition 10 *$\mathcal{F} \subseteq 2^{[n]}$ is s-extremal if and only if there exists a Sperner family $\mathcal{S}$ such that $\mathcal{F}$ is $\mathcal{S}$-extremal.*

PROOF: First suppose that $\mathcal{F} \subseteq 2^{[n]}$ is s-extremal, i.e $|\mathcal{F}| = |\mathrm{Sh}(\mathcal{F})|$. By Remark 6 we know that if we let $\mathcal{S}$ to be the collection of all minimal sets not shattered by $\mathcal{F}$, then $\mathrm{Sh}(\mathcal{F}) = 2^{[n]} \setminus \mathrm{Up}(\mathcal{S}) = \mathcal{H}(\mathcal{S})$. This implies $|\mathcal{F}| = |\mathcal{H}(\mathcal{S})|$ which together with the fact that the elements of $\mathcal{S}$ are not shattered gives that $\mathcal{F}$ is $\mathcal{S}$-extremal.

Now suppose that $\mathcal{F}$ is $\mathcal{S}$-extremal for some Sperner family $\mathcal{S}$, i.e. $\mathcal{F}$ shatters no element of $\mathcal{S}$ and $|\mathcal{F}| = |\mathcal{H}(\mathcal{S})|$. From the proof of Proposition 9 if follows that this is possible only if $\mathrm{Sh}(\mathcal{F}) = \mathcal{H}(\mathcal{S})$. However this means that $|\mathrm{Sh}(\mathcal{F})| = |\mathcal{H}(\mathcal{S})| = |\mathcal{F}|$ and so $\mathcal{F}$ is s-extremal. □

Let us now get back to families of the form $\mathcal{F}(\mathcal{S}, h)$. For $\mathcal{S} = \{S_1, \ldots, S_N\}$, to simplify notation, put $h(S_i) = H_i$. To analyse (2) in Corollary 8 first note that it holds if and only if $\mathrm{Up}(\mathcal{S}) = \bigcup_{i=1}^N \mathcal{P}_{S_i}$ and $2^{[n]} \setminus \mathcal{F}(\mathcal{S}, h) = \bigcup_{i=1}^N \mathcal{Q}_{S_i,H_i}$ have the same size. To study this we will use the inclusion-exclusion formula. For this note that $\mathcal{P}_{S_i} \cap \mathcal{P}_{S_j} = \mathcal{P}_{S_i \cup S_j}$ for every $1 \leq i < j \leq N$ and $\mathcal{Q}_{S_i,H_i} \cap \mathcal{Q}_{S_j,H_j} = \mathcal{Q}_{S_i \cup S_j, H_i \cup H_j}$ if $S_i \cap H_j = S_j \cap H_i$ and $\mathcal{Q}_{S_i,H_i} \cap \mathcal{Q}_{S_j,H_j} = \emptyset$ otherwise. In particular this means that for $I \subseteq [N]$ we have that

$$\left|\bigcap_{i \in I} \mathcal{P}_{S_i}\right| = \left|\mathcal{P}_{\bigcup_{i \in I} S_i}\right| = \left|\mathcal{Q}_{\bigcup_{i \in I} S_i, \bigcup_{i \in I} H_i}\right| = \left|\bigcap_{i \in I} \mathcal{Q}_{S_i,H_i}\right|$$

whenever $\bigcap_{i \in I} \mathcal{Q}_{S_i,H_i}$ is non-empty, which happens exactly if for every $i \neq j \in I$ we have $S_i \cap H_j = S_j \cap H_i$. Let $\mathbb{I}_{i,j}$ be the indicator of the event $S_i \cap H_j = S_j \cap H_i$, i.e. it is 1 if the equality is satisfied and 0 otherwise. As $\mathrm{Up}(\mathcal{S}) = \bigcup_{i=1}^N \mathcal{P}_{S_i}$ and $2^{[n]} \setminus \mathcal{F}(\mathcal{S}, h) = \bigcup_{i=1}^N \mathcal{Q}_{S_i,H_i}$, the inclusion-exclusion formula gives that we have $|\mathcal{H}(\mathcal{S})| = |\mathcal{F}(\mathcal{S}, h)|$ if and only if

$$\sum_{I \subseteq [N]} (-1)^{|I|+1} \left|\bigcap_{i \in I} \mathcal{P}_{S_i}\right| = \sum_{I \subseteq [N]} (-1)^{|I|+1} \left|\bigcap_{i \in I} \mathcal{Q}_{S_i,H_i}\right| = \sum_{I \subseteq [N]} (-1)^{|I|+1} \left(\prod_{i \neq j \in I} \mathbb{I}_{i,j}\right) \left|\bigcap_{i \in I} \mathcal{P}_{S_i}\right|$$

This latter equation can also be rewritten as

$$\sum_{I \subseteq [N]} (-1)^{|I|} \left(1 - \prod_{i \neq j \in I} \mathbb{I}_{i,j}\right) \left|\bigcap_{i \in I} \mathcal{P}_{S_i}\right| = 0$$

Proposition 11 *Let $\mathcal{S} = \{S_1, ..., S_N\} \subseteq 2^{[n]}$ be a Sperner family and $A \subseteq [n]$ be a fixed set. Furthermore let $h_A : \mathcal{S} \to 2^{[n]}$ be defined as $h_A(S) = S \cap A$, i.e. $H_i = h_A(S_i) = S_i \cap A$ for $i \in [N]$. Then $\mathcal{F}(\mathcal{S}, h_A)$ is s-extremal and $Sh(\mathcal{F}(\mathcal{S}, h_A)) = \mathcal{H}(\mathcal{S})$.*

PROOF: For the proof only note that in this case, for every $1 \leq i < j \leq N$ we have

$$S_j \cap H_i = S_j \cap S_i \cap A = S_i \cap S_j \cap A = S_i \cap H_j,$$

i.e. $\mathbb{I}_{i,j} = 1$. In this case $1 - \prod_{i \neq j \in I} \mathbb{I}_{i,j} = 0$ for every $I \subseteq [N]$, and so

$$\sum_{I \subseteq [N]} (-1)^{|I|} \left(1 - \prod_{i \neq j \in I} \mathbb{I}_{i,j}\right) \left|\bigcap_{i \in I} \mathcal{P}_{S_i}\right| = 0.$$

Equivalently this means that $|\mathcal{H}(\mathcal{S})| = |\mathcal{F}(\mathcal{S}, h)|$, and so by Corollary 8 $\mathcal{F}(\mathcal{S}, h_A)$ is s-extremal and $\mathrm{Sh}(\mathcal{F}(\mathcal{S}, h_A)) = \mathcal{H}(\mathcal{S})$. $\square$

Unfortunately, the converse is not true, i.e. if $\mathcal{F}(\mathcal{S}, h)$ is s-extremal and $\mathrm{Sh}(\mathcal{F}(\mathcal{S}, h)) = \mathcal{H}(\mathcal{S})$ then there does not necessarily exist a set $A \subseteq [n]$ such that $h = h_A$, as shown by the following example.

Example 12 *Let $n = 3$ and $\mathcal{S} = \{S_1, S_2, S_3\}$, where $S_1 = \{1, 2\}$, $S_2 = \{1, 3\}$ and $S_3 = \{2, 3\}$. Furthermore take h such that $H_1 = \{1\}$, $H_2 = \emptyset$ and $H_3 = \emptyset$. Then*

$$\mathcal{P}_1 = \mathcal{P}_{S_1} = \{\{1, 2\}, \{1, 2, 3\}\},\ \mathcal{P}_2 = \mathcal{P}_{S_2} = \{\{1, 3\}, \{1, 2, 3\}\},\ \mathcal{P}_3 = \mathcal{P}_{S_3} = \{\{2, 3\}, \{1, 2, 3\}\}$$

$$\mathcal{Q}_1 = \mathcal{Q}_{S_1, H_1} = \{\{1\}, \{1, 3\}\},\ \mathcal{Q}_2 = \mathcal{Q}_{S_2, H_2} = \{\emptyset, \{2\}\},\ \mathcal{Q}_3 = \mathcal{Q}_{S_3, H_3} = \{\emptyset, \{3\}\},$$

and so

$$\mathcal{F}(\mathcal{S}, h) = 2^{[3]} \setminus (\mathcal{Q}_1 \cup \mathcal{Q}_2 \cup \mathcal{Q}_3) = \{\{3\}, \{1, 2\}, \{2, 3\}, \{1, 2, 3\}\}.$$

On the other hand

$$\mathcal{H}(\mathcal{S}) = 2^{[3]} \setminus (\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3) = \{\emptyset, \{1\}, \{2\}, \{3\}\}$$

and so as both have size 4, by Corollary 8 $\mathcal{F}(\mathcal{S}, h)$ is s-extremal and $Sh(\mathcal{F}(\mathcal{S}, h)) = \mathcal{H}(\mathcal{S})$. However it is easily seen that there is no $A \subseteq [3]$ such that $h = h_A$ would hold.

Even if one cannot obtain every s-extremal family as $\mathcal{F}(\mathcal{S}, h_A)$ for some suitable $\mathcal{S}$ and function h_A, families of the above form seem to provide interesting examples of s-extremal families. For example setting $A = [n]$ we get back exactly the down-sets. What follows, as a main result we will show that Conjecture 4 holds for set systems of the form $\mathcal{F}(\mathcal{S}, h_A)$.

Theorem 13 *Let $A \subseteq [n]$ and let $\mathcal{S} \subseteq 2^{[n]}$ be a non-empty Sperner family. Then Conjecture 4 holds for $\mathcal{F}(\mathcal{S}, h_A)$, i.e. there is $F \notin \mathcal{F}(\mathcal{S}, h_A)$ such that $\mathcal{F}' = \mathcal{F}(\mathcal{S}, h_A) \cup \{F\}$ is again s-extremal. Moreover $\mathcal{F}' = \mathcal{F}(\mathcal{S}', h_A)$ for some suitable Sperner family $\mathcal{S}'$.*

PROOF: To shorten notation put $\mathcal{F} = \mathcal{F}(\mathcal{S}, h_A)$. Recall that by Proposition 11 $\mathcal{F}$ is s-extremal, i.e. $|\mathcal{F}| = |\mathrm{Sh}(\mathcal{F})|$ and $\mathrm{Sh}(\mathcal{F}) = \mathcal{H}(\mathcal{S})$. Pick an arbitrary $S_0 \in \mathcal{S}$ with $H_0 = S_0 \cap A$. Then there exists a unique (possibly empty) family $\{S_1', ..., S_k'\} \subseteq \{S_0 \cup \{v\} \ : \ v \in [n] \setminus S_0\}$ such that $\mathcal{S}' = (\mathcal{S} \setminus \{S_0\}) \cup \{S_1', ..., S_k'\}$ is again a Sperner familiy and

$$\mathcal{H}(\mathcal{S}') = 2^{[n]} \setminus \mathrm{Up}(\mathcal{S}') = \left(2^{[n]} \setminus \mathrm{Up}(\mathcal{S})\right) \cup \{S_0\} = \mathcal{H}(\mathcal{S}) \cup \{S_0\}.$$

For $i \in [k]$ let $H_i' = S_i' \cap A$ and let $\mathcal{F}'$ be the shorthand notation for $\mathcal{F}(\mathcal{S}', h_A)$. Again, by Proposition 11, $\mathcal{F}'$ is s-extremal and $\mathrm{Sh}(\mathcal{F}') = \mathcal{H}(\mathcal{S}')$. In particular, since $|\mathrm{Sh}(\mathcal{F}')| = |\mathcal{H}(\mathcal{S}')| = |\mathcal{H}(\mathcal{S}) \cup \{S_0\}| = |\mathcal{H}(\mathcal{S})| + 1 = |\mathrm{Sh}(\mathcal{F})| + 1$, we have $|\mathcal{F}'| = |\mathcal{F}| + 1$. Accordingly all that remains to be shown to prove the theorem is that $\mathcal{F} \subseteq \mathcal{F}'$, since in that case the unique set F in $\mathcal{F}' \setminus \mathcal{F}$ is a good choice. To see this, first note that $\mathcal{Q}_{S_i', H_i'} \subseteq \mathcal{Q}_{S_0, H_0}$ since $S_0 \subseteq S_i'$ for every $i \in [k]$, and hence

$$\bigcup_{i=1}^{k} \mathcal{Q}_{S_i', H_i'} \subseteq \mathcal{Q}_{S_0, H_0}.$$

However in this case

$$\mathcal{F} = \left(2^{[n]} \setminus \bigcup_{\substack{S\in\mathcal{S}\\ S\neq S_0}} \mathcal{Q}_{S,S\cap A}\right) \setminus \mathcal{Q}_{S_0,H_0} \subseteq \left(2^{[n]} \setminus \bigcup_{\substack{S\in\mathcal{S}\\ S\neq S_0}} \mathcal{Q}_{S,S\cap A}\right) \setminus \bigcup_{i=1}^{k} \mathcal{Q}_{S_i',H_i'} = 2^{[n]} \setminus \bigcup_{S\in\mathcal{S}'} \mathcal{Q}_{S,S\cap A} = \mathcal{F}',$$

as desired. □

4 Concluding remarks

Theorem 13 solves only a further special case of Conjecture 4, so the conjecture remains wide open in general. However the approach presented offers a possible way to approach it.

If we have a general s-extremal family $\mathcal{F} \subseteq 2^{[n]}$, then as noted earlier, there is a suitable Sperner family $\mathcal{S} \subseteq 2^{[n]}$ and a function $h : \mathcal{S} \to 2^{[n]}$ such that $\mathcal{F} = \mathcal{F}(\mathcal{S}, h)$. Similarly as in the proof of Theorem 13, we can take some $S_0 \in \mathcal{S}$ with $H_0 = h(S_0)$ and replace it with sets from $\{S_0 \cup \{v\} \ : \ v \in [n] \setminus S_0\}$ to obtain $\mathcal{S}'$ with $\mathcal{H}(\mathcal{S}') = \mathcal{H}(\mathcal{S}) \cup \{S_0\}$. To extend h from $\mathcal{S}$ to $\mathcal{S}'$, for each new set $S_0 \cup \{v\} \in \mathcal{S}' \setminus \mathcal{S}$ a reasonable choice for $h(S_0 \cup \{v\})$ is either H_0 or $H_0 \cup \{v\}$. In this case for $\mathcal{F}' = \mathcal{F}(\mathcal{S}', h)$ we will still have that $\mathcal{F} \subseteq \mathcal{F}'$ and $|\mathcal{F}'| \leq |\mathcal{F}| + 1$, however now $\mathcal{F} = \mathcal{F}'$ might be possible. Indeed, consider Example 12. If we take any $S_0 \in \mathcal{S}$, then one does not need to add any set to $\mathcal{S} \setminus \{S_0\}$, as we already have $\mathcal{H}(\mathcal{S}') = \mathcal{H}(\mathcal{S} \setminus \{S_0\}) = \mathcal{H}(\mathcal{S}) \cup \{S_0\}$. However if we were to choose $S_0 = S_3 = \{2, 3\}$, then the resulting $\mathcal{F}'$ is the the same as $\mathcal{F}$. In the special case, when $h = h_A$ for some $A \subseteq [n]$, this was not possible by the s-extremality of $\mathcal{F}'$, which was guaranteed by Proposition 11. Here we remark, that $\mathcal{F} = \mathcal{F}'$ does not contradict with the uniqueness of $\mathcal{S}$ and h suggested by Remark 6, as for $\mathcal{S}'$ we have that $\mathrm{Sh}(\mathcal{F}) \subsetneq \mathcal{H}(\mathcal{S}')$. In the above example for instance $\mathrm{Sh}(\mathcal{F}) = \mathcal{H}(\mathcal{S}) = \{\emptyset, \{1\}, \{2\}, \{3\}\} \subsetneq \{\emptyset, \{1\}, \{2\}, \{3\}, \{2, 3\}\} = \mathcal{H}(\mathcal{S}')$. Accordingly the main issue here is to rule out the possibility $\mathcal{F} = \mathcal{F}'$ by choosing S_0 and the new values for h carefully. Let us mention that in the above example S_1 and S_2 are good choices for S_0. Note that to prove the conjecture we need only one good instance. A possible step in this direction would be to characterize for a given Sperner family $\mathcal{S}$ the possible functions h such that $\mathcal{F}(\mathcal{S}, h)$ is s-extremal.

Another way to attack Conjecture 4 would be to try to adapt the proof from the case of maximum classes, i.e. $\binom{[n]}{k}$-extremal families to other $\mathcal{S}$-extremal systems, where $\mathcal{S}$ is a general Sperner family.

We find both approaches promising and are looking forward to working them out in more detail.

References

[1] W. W. Adams, P. Loustaunau, An Introduction to Gröbner bases, *Graduate Studies in Mathematics, Vol. 3*, American Mathematical Society (1994)

[2] R.P. Anstee, Properties of (0-1)-matrices with no triangles, *Journal of Combinatorial Theory, Series A* **29:186-198** (1980)

[3] R.P. Anstee, L. Rónyai, A. Sali, Shattering News, *Graphs and Combinatorics* **18:59-73** (2002)

[4] B. Bollobás, I. Leader, A.J. Radcliffe, Reverse Kleitman Inequalities, *Proceedings of the London Mathematical Society* **s3-58:153-168** (1989)

[5] B. Bollobás, A.J. Radcliffe, Defect Sauer Results, *Journal of Combinatorial Theory, Series A* **72:189-208** (1995)

[6] B. Chornomaz, Convex geometries are extremal for generalized Sauer-Shelah bound, **hal-01358594** (2016)

[7] P. Frankl, Extremal set systems, *In: R.L. Graham, M. Grtschel, L. Lovász (eds.), Handbook of Combinatorics Vol. 2*, MIT Press, Cambridge (1996)

[8] P. Frankl, J. Pach, On disjointly representable sets, *Combinatorica* **4:39-45** (1994)

[9] Z. Füredi, F. Quinn, Traces of finite sets, *Ars Combinatoria* **18:195-200** (1983)

[10] L. Kozma, S. Moran, Shattering, graph orientations and connectivity, *The Electronic Journal of Combintaorics* **20(3):P44** (2013)

[11] D. Kuzmin, M.K. Warmuth, Unlabelled compression schemes for maximum classes, *Journal of Machine Learning Research* **8(Sep):2047-2081** (2007)

[12] T. Mészáros, L. Rónyai, Shattering-Extremal Set Systems of Small VC-Dimnesion, *ISRN Combinatorics* **2013:126214** (2013)

[13] T. Mészáros, L. Rónyai, Shattering-extremal set systems of VC dimension at most 2, *The Electronic Journal of Combintorics* **21(4):P4.30** (2014)

[14] S. Moran, M.K. Warmuth, Labeled Compression Schemes for Extremal Classes, *In: R. Ortner, H. Simon, S. Zilles (eds.), Algorithmic Learning Theory, ALT 2016, Lecture Notes in Computer Science* **9925:33-49**, Springer (2016)

[15] L. Rónyai, T. Mészáros, Some combinatorial application of Gröbner bases, *In: F. Winkler (ed.), Algebraic Informatics, CAI 2011, Lecture Notes in Computer Science* **6742:65-83**, Springer (2011)

[16] B.I.P. Rubinstein, H. Rubinstein, A geometric approach to sample compression, *Journal of Machine Learning Research* **13(Apr):1221-1261** (2012)

[17] N. Sauer, On the density of families of sets, *Journal of Combinatorial Theory, Series A* **13:145-147** (1972)

[18] S. Shelah, A combinatorial problem: Stability and order for models in infinitary language, *Pacific Journal of Mathematics* **41:247-261** (1972)

[19] V.N. Vapnik, A.Y. Chervonenkis, On the Uniform Convergence of Relative Frequencies of Events to their Probabilities, *Theory of Probability and its Applications* **16:264-280** (1971)

On spanning trees with constraints on the leaf degree

Shun-ichi Maezawa

Graduate School of Systems Engineering and Science
Shibaura Institute of Technology
307 Fukasaku, Saitama, 337-8570 Japan
mf16061@shibaura-it.ac.jp

Ryota Matsubara[1]

Department of Mathematics
Shibaura Institute of Technology
307 Fukasaku, Saitama, 337-8570 Japan
ryota@sic.shibaura-it.ac.jp

Haruhide Matsuda[1]

Department of Mathematics
Shibaura Institute of Technology
307 Fukasaku, Saitama, 337-8570 Japan
hmatsuda@sic.shibaura-it.ac.jp

Abstract: Let T be a tree. The *leaf degree* of a vertex x in T is defined as the number of end-vertices in T adjacent to x. Let G be a graph and let f be an integer-valued function defined on $V(G)$ such that $f(x) \geq 0$ for all $x \in V(G)$. Then a tree T of G is said to be an *f-leaf-tree* of G if the leaf degree of each vertex $x \in V(T)$ is at most $f(x)$. For a positive integer m, an f-leaf-tree is an m-leaf-tree if $f(x) = m$ for any $x \in V(G)$. This paper shows a necessary and sufficient condition for graphs to have a spanning f-leaf-tree.

Keywords: Graph; Factor; Tree; Leaf

1 Introduction

We consider a finite undirected graph with neither loops nor multiple edges. For a graph G, we denote the vertex set of G by $V(G)$ and the edge set of G by $E(G)$. The order of G is denoted by $|G|$.

Let K_m denote a complete graph of order m, C_m a cycle of order m, and $K_{1,m}$ a star with m end-vertices. For a set of undirected graphs $\mathcal{H}$, a spanning subgraph F of a graph G is called an $\mathcal{H}$-factor if every connected component of F is isomorphic to some element of $\mathcal{H}$. If $\mathcal{H} = \{K_2\}$, then we obtain Tutte's characterization on the existence of perfect matchings [4]. Tutte also shows the following result, where $i(G)$ stands for the number of isolated vertices in G.

Theorem 1 (Tutte) *A graph G has a $\{K_2, C_i \mid i \geq 3\}$-factor if and only if*

$$i(G - S) \leq |S| \quad \text{for all } S \subseteq V(G).$$

On the other hand, Amahashi and Kano consider the characterization for a graph to have stars.

Theorem 2 (Amahashi and Kano [1]) *Let $n \geq 2$ be an integer. A graph G has a $\{K_{1,1}, K_{1,2}, \ldots, K_{1,n}\}$-factor if and only if*

$$i(G - S) \leq n|S| \quad \text{for all } S \subseteq V(G).$$

[1]This work was supported by JSPS KAKENHI, Grant-in-Aid for Scientific Research(C), Grant Number 15K04980.

Let G be a graph and f an integer-valued function defined on $V(G)$ such that $f(x) \geq 1$ for all $x \in V(G)$. Then a subgraph of G is called an *f-star* if it is the star $K_{1,t}$ whose degree t of the center x satisfies $1 \leq t \leq f(x)$. An *f-star factor* of G is a spanning subgraph each of whose components is an f-star.

Theorem 3 (Berge and Las Vergnas [2]) *Let G be a graph and f an integer-valued function defined on $V(G)$ such that $f(x) \geq 2$ for all $x \in V(G)$. Then G has an f-star factor if and only if*

$$i(G - S) \leq \sum_{x \in S} f(x) \quad \text{for all } S \subseteq V(G).$$

Theorem 4 (Berge and Las Vergnas [2]) *Let G be a graph and f an integer-valued function defined on $V(G)$ such that $f(x) \geq 1$ for all $x \in V(G)$. Then G has a spanning subgraph of G every component of which is either an f-star, or an odd cycle with $f(x) = 1$ for every vertex x if and only if*

$$i(G - S) \leq \sum_{x \in S} f(x) \quad \text{for all } S \subseteq V(G).$$

For a tree T, the *leaf degree* of a vertex $x \in V(T)$ is defined as the number of end-vertices in T adjacent to x, and is denoted by $\mathrm{leaf}_T(x)$.

Let f be an integer-valued function defined on $V(G)$ such that $f(x) \geq 0$ for all $x \in V(G)$. Then a tree T of G is said to be an *f-leaf-tree* of G if the leaf degree of each vertex $x \in V(T)$ is at most $f(x)$. For a positive integer m, an f-leaf-tree is an m-leaf-tree if $f(x) = m$ for any $x \in V(G)$.

On the 1st Japanese-Hungarian Symposium for Discrete Mathematics and its Applications (Kyoto, 1999), Kaneko presented a good criterion for a graph to have a spanning m-leaf-tree.

Theorem 5 (Kaneko [3]) *Let m be an integer with $m \geq 1$. A connected graph G has a spanning m-leaf-tree if and only if for every nonempty subset $S \subseteq V(G)$,*

$$i(G - S) \leq (m + 1)|S| - 1$$

unless G is isomorophic to K_3 and $m = 1$.

2 Main result

Motivated by Kaneko's theorem, we have a necessary and sufficient condition for a graph to have a spanning f-leaf-tree. In particular, our theorem contains the case when $f(x) = 0$ for some vertices $x \in V(G)$.

Theorem 6 *Let G be a connected graph and f an integer-valued function defined on $V(G)$ such that $f(x) \geq 0$ for all $x \in V(G)$. Suppose that the set of vertices x with $f(x) = 0$ is independent in G. Then G has a spanning f-leaf-tree if and only if for every nonempty subset $S \subseteq V(G)$,*

$$i(G - S) \leq \sum_{x \in S} (f(x) + 1) - 1$$

unless G is isomorophic to K_3 such that $f(x_i) = 1$ for $x_i \in V(K_3)$ with $i = 1, 2$ and $f(x_3) \leq 1$.

To show the proof, we need some notation.

For a vertex $x \in V(G)$, we denote by $N_G(x)$ the set of vertices adjacent to x in G. For a subset $X \subseteq V(G)$, write $N_G(X) = \bigcup_{x \in X} N_G(x)$. For a subset S of $V(G)$, denote by $G[S]$ the subgraph of G induced by the vertex set S, i.e., the graph having vertex set S and whose edge set consists of those edges of G incident with two vertices of S. In particular, if S is a subgraph of G, then we simply write $G[S]$ for $G[V(S)]$.

A graph H is said to be a *triangle-tree* if it satisfies the following two conditions:

(i) H is a tree or a connected graph such that every cycle of H is K_3, i.e., every block of H is an edge or a cycle with three vertices.

(ii) No two cycles of H have a vertex in common.

Note that K_3 is also a triangle-tree. In the proof below, the number of end-vertices in a triangle-tree T adjacent to a vertex $x \in V(T)$ is also called the *leaf degree* of a vertex x and is denoted by $\mathrm{leaf}_T(x)$.

PROOF: We first prove necessity. Let T be a spanning tree of G satisfying $\mathrm{leaf}_T(x) \leq f(x)$ for any $x \in V(T)$. Suppose, to the contrary, that there exists nonempty subset $S \subseteq V(G)$ such that $i(G-S) \geq \sum_{x\in S}(f(x)+1)$. Let W be the set of isolated vertices in $V(G)-S$ whose degree in T is at least two. Then $i(G-S)-|W|$ is the number of leaves of T in $I(G-S)$, where $I(G-S)$ denotes the set of isolated vertices in $G-S$. On the other hand, by $N_G(I(G-S)) \subseteq S$ and $\mathrm{leaf}_T(x) \leq f(x)$ for all $x \in V(G)$, the number of leaves of T in $I(G-S)$ is at most $\sum_{x\in S} f(x)$. Therefore

$$\sum_{x\in S} f(x) \geq i(G-S) - |W| \geq \sum_{x\in S}(f(x)+1) - |W|,$$

which implies $|W| \geq |S| \geq 1$. Since $\mathrm{leaf}_T(x) \geq 2$ for any $x \in W$, the induced subgraph $T[W \cup S]$ contains at least $2|W|$ edges, which implies $T[W \cup S]$ is a forest having at least $2|W| \geq |W| + |S|$ edges. This is a contradiction. Hence $i(G-S) \leq \sum_{x\in S}(f(x)+1) - 1$ for all $\emptyset \neq S \subseteq V(G)$.

We next show the sufficiency. For two subgraph H and X of G (possibly $V(H) = \emptyset$ or $V(X) = \emptyset$), a pair (H, X) is called *f-admissible* if the pair satisfies the following four conditions:

(i) $V(H) \cap V(X) = \emptyset$ and $V(H) \cup V(X) = V(G)$,

(ii) each component of X is isomorphic to $K_{1,f(x)+1}$ for the center x,

(iii) the order of each component in H is at least two, and

(iv) each component T of H is either a tree with $\mathrm{leaf}_H(x) \leq f(x)$ for all $x \in V(T)$, or a triangle-tree with $\mathrm{leaf}_H(x) \leq 1$ for all $x \in V(T)$ such that any non-leaf vertex with leaf degree 0 in H is contained in K_3 in H.

We now show the existence of an f-admssible pair of G. Since for every nonempty subset $S \subseteq V(G)$,

$$i(G-S) \leq \sum_{x\in S}(f(x)+1) - 1 < \sum_{x\in S}(f(x)+1),$$

by Theorem 4, G has a spanning subgraph F every component of which is either an $(f+1)$-star, or an odd cycle with $f(x) = 0$ for every vertex x. By the assumption of Theorem 6, F contains no odd cycles. Define X as the set of all components $K_{1,f(x)+1}$ and let $H = F - V(X)$. Then this pair (H, X) is an f-admissible pair of G.

Choose an f-admissible pair (H, X) of G such that $|V(H)|$ is as large as possible.

Claim 7 $X = \emptyset$.

Proof. Suppose, to the contrary, that $X \neq \emptyset$. Let $K_{1,f(s_1)+1}, \ldots, K_{1,f(s_k)+1}$ be the components of X for each $i = 1, \ldots, k$, where s_i is the center of $K_{1,f(s_i)+1}$ and let $S = \{s_1, \ldots, s_k\}$. Then $X - S$ consists of $\sum_{i=1}^{k}(f(s_i)+1)$ isolated vertices. On the other hand, we have $i(G-S) \leq \sum_{x\in S}(f(x)+1) - 1$ for all $\emptyset \neq S \subseteq V(G)$ by the assumption of Theorem 6. Thus a vertex $x \in V(X) - S$ is adjacent to a vertex $y \in V(H) \cup (V(X) - (S \cup \{x\}))$ in G. Without loss of generality, we may assume that $x \in V(K_{1,f(s_1)+1}) - \{s_1\}$. Note that $f(x) > 0$ or $f(s_1) > 0$ holds by the assumption that two vertices x and y with $f(x) = f(y) = 0$ are independent in G.

To deduce a contradiction, we distinguish two cases.

Case 1 $y \in V(H)$.

Let C be the component of H containing y. Then $C \cup xy \cup K_{1,f(s_1)+1}$ is a triangle-tree such that $\mathrm{leaf}_T(v) \leq f(v)$ for all $v \in V(T)$. In particular, if C is a triangle-tree containing cycles, then $C \cup xy \cup K_{1,f(s_1)+1}$ is also a triangle-tree containing cycles since the pair (H, X) is an f-admissible pair of G, $\mathrm{leaf}_C(z) = 0$ for any vertex z contained in a cycle in C, and thus $\mathrm{leaf}_{C\cup xy\cup K_{1,f(s_1)+1}}(z) = 0$ for any vertex z contained in a cycle in $C \cup xy \cup K_{1,f(s_1)+1}$. Moreover, if $f(x) = 0$, then $f(s_1) > 0$ by the asumption of the theorem and hence s_1 is not a leaf of $C\cup xy\cup K_{1,f(s_1)+1}$. Let $H' = H - \{C\} + \{C\cup xy\cup K_{1,f(s_1)+1}\}$ and $X' = X - \{K_{1,f(s_1)+1}\}$. Then the pair (H', X') is an f-admissible pair of G such that $|V(H)| < |V(H')|$, which contradicts the maximal property of H.

Case 2 $y \in V(X) - (S \cup \{x\})$.

Suppose first that $y \in V(K_{1,f(s_i)+1})$ for some $i = 2, \ldots, k$. Then $y \neq s_i$ and $K_{1,f(s_1)+1}\cup xy\cup K_{1,f(s_i)+1}$ is a tree such that the leaf degree of each vertex v is at most $f(v)$. Let $H' = H + \{K_{1,f(s_1)+1} \cup xy \cup K_{1,f(s_i)+1}\}$ and $X' = X - \{K_{1,f(s_1)+1}, K_{1,f(s_i)+1}\}$. Then the pair (H', X') is an f-admissible pair of G such that $|V(H)| < |V(H')|$, which contradicts the maximal property of H.

Hence we consider the case when $y \in V(K_{1,f(s_1)+1}) - \{x, s_1\}$. Note that $f(x) > 0$ or $f(y) > 0$ holds by the assumption of the theorem. By symmetry, we may assume that $f(x) > 0$. If $f(s_1) \geq 2$, then $K_{1,f(s_1)+1} + xy - s_1y$ is a tree such that the leaf degree of each vertex v is at most $f(v)$. Put $H' = H + \{K_{1,f(s_1)+1} + xy - s_1y\}$.

If $f(s_1) = 1$, then $K_{1,f(s_1)+1}\cup xy$ is a cycle with three vertices, i.e., K_3. Put $H' = H+\{K_{1,f(s_1)+1}\cup xy\}$. In either case, setting $X' = X - V(K_{1,f(s_1)+1})$, we obtain that the pair (H', X) is an f-admissible pair of G such that $|V(H)| < |V(H')|$, which contradicts the maximal choice of H.

Thus $V(X) = \emptyset$ and H is a spanning subgraph of G, as claimed.

We construct the desired spanning tree. Contract each component C of H by a vertex. The resulting graph is connected and thus has a spanning tree T' since G is connected. Then H together with all edges of T' is a spanning triangle-tree T such that $\mathrm{leaf}_{T'}(x) \leq f(x)$ for all $x \in V(H)$. If C contains K_3, then no K_3 contains two vertices v with $f(v) = 0$. By deleting a suitable edge from each K_3 of H, we obtain a desired spanning tree of G. The proof is completed. □

3 Remark

In Theorem 6, the assumption "the set of vertices x with $f(x) = 0$ is independent in G" is necessary. This is shown in the following graph G:

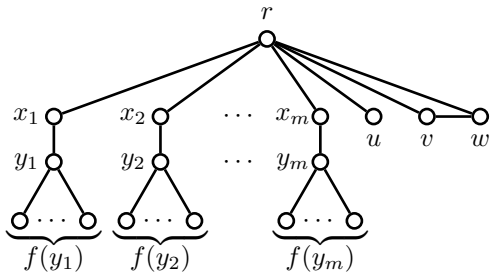


Fig.1 the graph G

Let m be a positive integer and let f be an integer-valued function defined on $V(G)$ such that $f(r) = 1$, $f(x_i) = 1$ for each $x_i \in V(G)$ with $i = 1, 2, \ldots, m$, and $f(x) = 0$ for all vertex $x \in V(G) \setminus \{r, x_i, y_i\}$ with $i = 1, 2, \ldots, m$. For each $y_i \in V(G)$ with $i = 1, 2, \ldots, m$, define $f(y_i)$ as any non-negative integer.

The above graph G satisfies the condition $i(G - S) \leq \sum_{x\in S}(f(x) + 1) - 1$ for all non-empty subset $S \subseteq V(G)$, and contains two adjacent vertices v, w with $f(v) = f(w) = 0$.

Then G has no spanning f-leaf-tree. In fact, G has three distinct spanning trees $T_1 = G - rv$, $T_2 = G - rw$, and $T_3 = G - vw$, however, $\mathrm{leaf}_{T_1}(w) = 1 > f(w) = 0$, $\mathrm{leaf}_{T_2}(v) = 1 > f(v) = 0$, and $\mathrm{leaf}_{T_3}(r) = 3 > f(r) = 1$. Each spanning tree T_i of G does not satisfy the definition of a spanning f-leaf-tree, and thus G has no desired spanning tree.

4 Acknowledgement

The authors would like to thank the referees for their helpful comments and suggestions.

References

[1] A. Amahashi and M. Kano, Factors with given components, *Discrete Mathematics* **42** (1982) 1–6.

[2] C. Berge and M. Las Vergnas, On the existence of subgraphs with degree constraints, *Nederl. Akad. Wetensch. Indag. Math.* **40** (1978) 165–176.

[3] A. Kaneko, Spanning trees with constraints on the leaf degree, *Discrete Applied Mathematics* **115** (2001) 73–76.

[4] W. T. Tutte, The factorization of linear graphs, *J. London Math. Soc.* **22** (1947) 107–111.

Making Bidirected Graphs Strongly Connected

Tatsuya Matsuoka[1]

Department of Mathematical Informatics
The University of Tokyo
Tokyo 113-8656, Japan
tatsuya_matsuoka@mist.i.u-tokyo.ac.jp

Shun Sato[1]

Department of Mathematical Informatics
The University of Tokyo
Tokyo 113-8656, Japan
shun_sato@mist.i.u-tokyo.ac.jp

Abstract: We consider problems to make a given bidirected graph strongly connected with minimum cardinality of additional signs or additional arcs. For the former problem, we show the minimum number of additional signs and give a linear-time algorithm for finding an optimal solution. For the latter problem, we give a linear-time algorithm for finding a feasible solution whose size is equal to the obvious lower bound or more than that by one.

Keywords: bidirected graph, strongly connected, condensation

1 Introduction

Problems to make a given graph (strongly) connected are well-investigated. The minimum number of additional edges to make a given undirected graph connected and that of additional arcs to make a given directed graph strongly connected [6] are well-known.

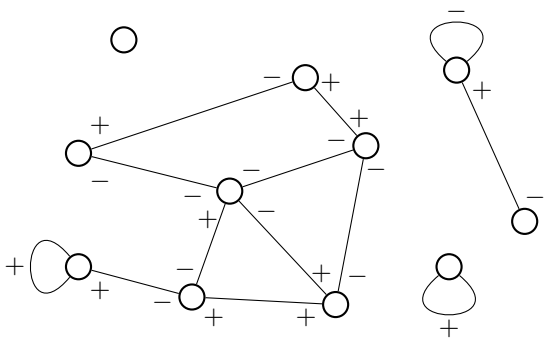


Figure 1: Bidirected Graph.

The concept of bidirected graphs (Fig. 1; the precise definition will be given later in Section 2) was introduced by Edmonds and Johnson [5]. It is a common generalization of undirected graphs and directed graphs. For bidirected graphs, Ando, Fujishige and Nemoto [3] defined the notion of strong connectivity and gave a linear-time algorithm for the strongly connected component decomposition. However, problems to make a given bidirected graph strongly connected have not been formulated.

In this paper, we consider problems to make a given bidirected graph strongly connected with minimum cardinality of additional signs or additional arcs.

[1]The authors are supported by JSPS Fellowship for Young Scientists.

1.1 Related Works

It is obvious that the minimum number of additional edges to make a given undirected graph connected is fewer than the number of connected components of a given graph by one. Eswaran and Tarjan [6] showed the minimum number of additional arcs to make a given directed graph strongly connected and that of additional edges to make a given undirected graph *bridge-connected* (*2-edge-connected*) or *biconnected* (*2-vertex-connected*). Linear-time algorithms for finding an optimal solution of these problems are also given in [6]. Note that they defined an operation called "condensation" which transforms a general directed graph to an acyclic directed graph. We can focus on the acyclic case since we can obtain a solution of the original problem by solving the problem on the condensed graph. For a directed graph $G = (V, A)$, $v \in V$ is a *source* if $\delta(v) \geq 1,\ \rho(v) = 0$, a *sink* if $\rho(v) \geq 1,\ \delta(v) = 0$ and an isolated vertex if $\rho(v) = \delta(v) = 0$ (in directed graphs, δ and ρ denote the out-degree and in-degree functions, respectively).

Theorem 1 (Eswaran–Tarjan [6]) *Let $G = (V, A)$ be an acyclic directed graph with the set $S \subseteq V$ of sources, the set $T \subseteq V$ of sinks and the set $Q \subseteq V$ of isolated vertices ($|S| + |T| + |Q| > 1$). Then the minimum number of additional arcs to make the given graph strongly connected is* $\max\{|S|, |T|\} + |Q|$.

For an undirected graph $G = (V, E)$, $v \in V$ is called a *pendant* if $\delta(v) = 1$ and $V' \subseteq V$ is called a *pendant block* if it is a 2-vertex-connected component and it contains exactly one *cutnode* (for undirected graphs, δ denotes the degree function). Note that $v \in V$ is a cutnode if the original graph is connected and the induced graph of $V \setminus \{v\}$ is disconnected. Similarly, $V' \subseteq V$ is called an *isolated block* if it is a 2-vertex-connected component and it contains no cutnode.

Theorem 2 (Eswaran–Tarjan [6]) *Let $G = (V, E)$ be an undirected graph with the set $P \subseteq V$ of pendants and the set $Q \subseteq V$ of isolated vertices ($|P| + |Q| > 1$). Then the minimum number of additional edges to make the given graph 2-edge-connected is $\lceil |P|/2 \rceil + |Q|$.*

Theorem 3 (Eswaran–Tarjan [6]) *Let $G = (V, E)$ be an undirected graph with the set $\mathcal{P} \subseteq 2^V$ of pendant blocks and the set $\mathcal{Q} \subseteq 2^V$ of isolated blocks ($|\mathcal{P}| + |\mathcal{Q}| > 1$), the minimum number of additional edges to make the given graph 2-vertex-connected is* $\max\{d - 1, \lceil |\mathcal{P}|/2 \rceil + |\mathcal{Q}|\}$. *Here,*

$$d := \max\{\#(\textit{2-vertex-connected components containing } v) + \#(\textit{connected components}) - 1 \mid v \in V\}$$

On the other hand, problems on bidirected graphs also have been considered in the literature. Ando, Fujishige and Nemoto [3] gave a linear-time algorithm for strongly connected component decomposition of bidirected graphs. This algorithm is made use of for the block triangularization of skew-symmetric matrices [11].

The strongly connected component decomposition of a bidirected graph [3] is obtained by the ordinary strongly connected component decomposition of the associated directed graph, *skew-symmetric graph*, which will be used in Section 3. As pointed out in [3], the same graph is used by Zaslavsky [14] for the study of *signed graphs* [10]. The notion of skew-symmetric graphs is defined firstly by Tutte [13] with the name "antisymmetrical digraphs" independent from bidirected graphs. There are also various problems on skew-symmetric graphs, and they have been intensively studied [7, 8, 9]. Study on bisubmodular polyhedra also made use of this skew-symmetric graph [2] with the name "exchangeability graph."

1.2 Our Contribution

In this paper, we formulate problems to make a given bidirected graph strongly connected with minimum cardinality of additional signs or additional arcs. Since self-loops have significance for the strong connectivity on bidirected graphs, these two problems arise depending on how to treat self-loops.

We first define the procedure called "condensation" on bidirected graphs. We can reduce general cases to acyclic cases by this operation for the above two problem settings. This can be done by using the strongly connected component decomposition algorithm for bidirected graphs devised by Ando, Fujishige and Nemoto [3]. This is similar to the fact that the condensation on directed graphs is done by using

strongly connected component decomposition of directed graphs [6, Lemma 1]. However, since there are signs for each arc in bidirected graphs, we must define the appropriate signs for each arc on the condensed bidirected graph and the procedure itself and validity are more nontrivial than the case of directed graphs.

We then discuss the two versions of the problems on bidirected graphs. For the problem on signs, the obvious lower bound can be obtained from the necessity for connectivity of the underlying graph and a condition on signs with each vertex. We show that this lower bound can be achieved for any acyclic bidirected graph and give a linear-time algorithm for finding an optimal solution. For the problem on arcs, we give a linear-time algorithm for finding a feasible solution whose size is equal to the obvious lower bound or more than that by one.

1.3 Organization

The organization of the rest of this paper is as follows. We give definitions and notation in Section 2. In Section 3, we give two problem settings dealt with in this paper and devise the condensation operation on bidirected graphs, which reduces a general case to an acyclic case. These two problems are discussed in Sections 4 and 5, respectively. Section 6 is devoted to concluding remarks involving other problem settings.

2 Preliminaries

In this section, we introduce definitions and notation used in this paper.

Definitions in this section mainly refer Ando and Fujishige [1] and Ando, Fujishige and Nemoto [3]. A *bidirected graph* is a triplet of a vertex set V, an arc set A and a boundary operator $\partial : A \to 3^V (:= \{(X, Y) \mid X, Y \subseteq V, X \cap Y = \emptyset\})$ such that $\partial a = (X_a, Y_a)$ satisfies $1 \leq |X_a| + |Y_a| \leq 2$ for each $a \in A$. We use the notation $|\partial a| := |X_a| + |Y_a|$. Let $\partial^+ : A \to 2^V$ and $\partial^- : A \to 2^V$ denote the operators with $\partial^+ a = X_a$ and $\partial^- a = Y_a$. This can be regarded that the signs are put on endpoints of *links* or on *self-loops* by ∂^+ and ∂^- (here we call an arc a link if it connects two different vertices). In other words, $\partial^+ a$ and $\partial^- a$ are the sets of endpoints of a with the signs "+" and "−", respectively. We call a vertex $v \in V$ an endpoint of $a \in A$ if $v \in \partial^+ a \cup \partial^- a$. We call an arc a with $\partial a = (\{v\}, \emptyset)$ a plus-loop at v and a with $\partial a = (\emptyset, \{v\})$ a minus-loop at v.

For brevity, we define some other notation. Let $\bar{\partial} : A \to 2^V$ denote the operator with $a \mapsto \partial^+ a \cup \partial^- a$ for each $a \in A$. For a bidirected graph $G = (V, A; \partial)$, let $\bar{G} = (V, A)$ be the undirected graph ignoring the signs of G (the *underlying graph* of G). We write as $a = (u, v)$ if $\bar{\partial} a = \{u, v\}$. Let us define a sign operator $\pi : \{(a, u) \mid a \in A, u \in \bar{\partial} a\} \to \{+, -\}$ as $\pi(a, u) = +$ if $u \in \partial^+ a$ and $\pi(a, u) = -$ if $u \in \partial^- a$. Let "(u, v) with (π_1, π_2)" $(\pi_1, \pi_2 \in \{+, -\})$ denotes an arc $a = (u, v)$ with $\pi(a, u) = \pi_1$ and $\pi(a, v) = \pi_2$.

An arc $a \in A$ is said to be *positively (negatively) incident to* v if $v \in \partial^+ a$ $(v \in \partial^- a)$. Arcs $a \in A$ and $a' \in A$ are said to be *oppositely incident to* v if a is positively (negatively) incident to v and a' is negatively (positively) incident to v.

An alternating sequence of vertices and arcs $(v_0, a_1, v_1, a_2, \ldots, a_l, v_l)$ $(l \geq 1)$ is called a *path* if a_i and a_{i+1} are oppositely incident to v_i $(i = 1, 2, \ldots, l-1)$, a_1 is incident to v_0 and a_l is incident to v_l. This is called $(\pi(a_1, v_0), \pi(a_l, v_l))$-path from v_0 to v_l. A path with $v_0 = v_l$ is called a *cycle* with a *root* $v_0 (= v_l)$. If a_l and a_1 are oppositely incident to v_0 and it includes distinct vertices, we call it a *proper* cycle. A cycle which is not proper is called an *improper* cycle. If a graph does not contain a proper cycle, we call it an *acyclic* graph (Note that this definition is different from that of "strongly acyclic" or "weakly (node- or edge-) acyclic" in [4]).

For a bidirected graph $G = (V, A; \partial)$, two vertices $v, v' \in V$ are called to be *strongly connected* if G contains two paths $(v, a_1^1, v_1^1, a_2^1, \ldots, a_{l_1}^1, v')$ and $(v, a_1^2, v_1^2, a_2^2, \ldots, a_{l_2}^2, v')$ such that a_1^1 and a_1^2 are oppositely connected to v and $a_{l_1}^1$ and $a_{l_2}^2$ are oppositely connected to v'. Note that these two paths need not to be vertex-disjoint. The binary relation on V can be defined by this strong connectivity: $v \sim v'$ if v and v' are strongly connected. By assuming that $v \sim v$ $(\forall v \in V)$, we obtain the equivalence relation $\sim$ on V. Each equivalence class of V on $\sim$ is called *strongly connected component* and G is called *strongly connected* if G has only one strongly connected component.

A vertex $v \in V$ is called *inconsistent* if there exist improper cycles $C_1 = (v, a_1^1, v_1^1, a_2^1, \ldots, a_{l_1}^1, v)$ and $C_2 = (v, a_1^2, v_1^2, a_2^2, \ldots, a_{l_2}^2, v)$ with root v such that $a_{l_2}^2$ and a_1^1 are oppositely incident to v. It is stated in [3] that if u and v are strongly connected and u is inconsistent, then v is also inconsistent. Thus, the notion of inconsistency can also be naturally defined on strongly connected components.

3 Settings and the Condensation Operation

In this section, we introduce the problem settings we tackle in this paper and explain the operation called condensation.

3.1 Problem Settings

We deal with the problems of the following type.

Problem 4 *Let $G = (V, A; \partial)$ be a bidirected graph. Find additional arcs A' and a boundary operator $\partial' : A \cup A' \to 3^V (\partial' a = \partial a \ (\forall a \in A))$ minimizing $F(A', \partial') := \sum_{a' \in A'} f(\partial' a')$ $(f : \{(X, Y) \mid X, Y \subseteq V, X \cap Y = \emptyset, 1 \leq |X| + |Y| \leq 2\} \to \mathbb{R})$ such that $G' := (V, A \cup A'; \partial')$ is a strongly connected bidirected graph.*

Note that Problem 4 is NP-hard in general. This can easily be shown by following the argument in the proof of [6, Theorem 1] as follows: we show this by reducing the following directed Hamiltonian cycle problem to Problem 4 with a certain function f.

Problem 5 (Directed Hamiltonian Cycle Problem) *Let $D = (V, A)$ be a directed graph. Find a directed Hamiltonian cycle in D.*

Set $f(\partial' a') = 1$ if $a' = (v_1, v_2)$ with $(+, -)$ and there exists $a = (v_1, v_2)$ in D and set $f(\partial' a') = 2$ for any other possible arc a'. There exists a solution of Problem 4 with respect to $G = (V, \emptyset; \partial)$ satisfying $F(A', \partial') = |V|$ if and only if there exists a solution of Problem 5. Since Problem 5 is NP-complete [12], Problem 4 is NP-hard.

For the problem on undirected graphs or directed graphs similar to Problem 4, it is natural to define $F(A', \partial') := |A'|$, i.e., minimization of the cardinality of additional edge (or arc) set. For bidirected graphs, however, there are two reasonable candidates of $F(A', \partial')$, i.e., $\sum_{a' \in A'} |\partial' a'|$ and $|A'|$. In other words, $f(\partial' a') := |\partial' a'|$ in the former setting and $f(\partial' a') := 1$ in the latter setting. The former means the minimization of the number of the additional signs on arcs and the latter means that of arcs themselves. In other words, the cost of a link is twice higher than that of a self-loop for the former problem and is the same for the latter problem. These natural two problems arise because self-loops have influence on strong connectivity in bidirected graphs (see, e.g., Fig. 2). Note that self-loops do not have any influence on the structure of (strong) connectivity for undirected graphs or directed graphs.

3.2 Reduction to Acyclic Case

We present a technique for reducing general cases to acyclic cases for Problem 4 with respect to $F(A', \partial') = \sum_{a' \in A'} |\partial' a'|$ or $F(A', \partial') = |A'|$.

For directed graphs, Eswaran and Tarjan [6] firstly condense the given directed graph to focus on acyclic cases. There, the condensed graph $\tilde{G} = (\tilde{V}, \tilde{A})$ is obtained from the strongly connected component decomposition $C_1, C_2, \ldots, C_k$ of the original graph $G = (V, A)$, where

$$\tilde{V} := \{v_{C_1}, v_{C_2}, \ldots, v_{C_k}\}, \qquad \tilde{A} := \left\{(v_{C_j}, v_{C_k}) \mid \exists v \in V(C_j), \exists v' \in V(C_k) \text{ s.t. } (v, v') \in A\right\}.$$

For bidirected graphs, however, due to the presence of the signs on arcs, the procedure of condensation becomes rather nontrivial. Fortunately, however, we can utilize the linear-time algorithm for strongly connected component decomposition devised by Ando, Fujishige and Nemoto [3]. Precisely speaking,

in order to appropriately define signs in the condensed graph, we use the strongly connected component decomposition of the associated skew-symmetric graph, which is almost equivalent to the strongly connected component decomposition of the original bidirected graph $G = (V, A; \partial)$ [3, Corollary 5.4].

In Algorithm CONDENSE(G) described below, steps 1–4 are the same as steps of the strongly connected component decomposition algorithm of [3]. For simplicity, the precise definition of the boundary operator $\tilde{\partial}$ is omitted and we alternatively write such as "add $\tilde{a}$ to $\tilde{A}$ with the sign $(+,-)$" (this notation is also employed in the sequel).

Algorithm CONDENSE(G)

Step 1 $\tilde{A} := \emptyset$.

Step 2 Construct the associated skew-symmetric graph $G^{\pm} = (V^+ \cup V^-, A^+ \cup A^-)$, where V^+ and V^- are copies of V ($v^+ \in V^+$ and $v^- \in V^-$ denote the copy of $v \in V$) and A^+ and A^- are defined as

$$A^+ = \{(v^{\pi(a,v)}, w^{-\pi(a,w)}) \mid a \in A, \bar{\partial} a = \{v, w\}\},\ A^- = \{(w^{\pi(a,w)}, v^{-\pi(a,v)}) \mid a \in A, \bar{\partial} a = \{v, w\}\}.$$

Note that v can be equal to w. Here, $-\pi$ is equal to $-$ (resp. $+$) if $\pi = +$ (resp. $-$).

Step 3 Decompose $G^{\pm}$ into strongly connected components $G^{\pm}_j = (U^{\pm}_j, B^{\pm}_j)$ $(j \in J)$.

Step 4 For each $j \in J$, define

$$U_j = \{v \in V \mid v^+ \in U^{\pm}_j\} \cup \{v \in V \mid v^- \in U^{\pm}_j\}.$$

Then, define W_i $(i \in I)$ be the distinct members of U_j $(j \in J)$ and partition I into I_1 and I_2 so that W_i appears twice (resp. once) in the family $\{U_j \mid j \in J\}$ for each $i \in I_1$ (resp. I_2).

Step 5 Define $\tilde{V} := \{\tilde{v}_i \mid i \in I\}$, here $\tilde{v}_i$ is the vertex corresponding to W_i.

Step 6 For each $i \in I$, choose a vertex $w_i \in W_i$. Define the map $\sigma : V \to \{+,-\}$ as $\sigma(v) = +$ if there exist $i \in I$ and $j \in J$ such that $w^+_i, v^+ \in U^{\pm}_j$ and $\sigma(v) = -$ otherwise.

Step 7 Partition I_1 into I^0_1, I^+_1 and I^-_1 so that w^-_i (resp. w^+_i) is reachable from w^+_i (resp. w^-_i) for each $i \in I^+_1$ (resp. $i \in I^-_1$) on the induced subgraph with respect to $\{v^+ \mid v \in W_i\} \cup \{v^- \mid v \in W_i\}$. (Both reachability are not attained for each element $\{w^+_i, w^-_i\}$ with $i \in I^0_1$.) Add a plus-loop at $\tilde{v}_i$ to $\tilde{A}$ for each $i \in I^+_1 \cup I_2$ and add a minus-loop at $\tilde{v}_i$ to $\tilde{A}$ for each $i \in I^-_1 \cup I_2$.

Step 8 For each pair $i_1, i_2 \in I$ $(i_1 \neq i_2)$, if there exists an arc $a \in A$ such that $\{u\} = \bar{\partial} a \cap W_{i_1}$ and $\{v\} = \bar{\partial} a \cap W_{i_2}$, then add an arc $\tilde{a} = (\tilde{v}_{i_1}, \tilde{v}_{i_2})$ to $\tilde{A}$ with the sign $(\sigma(u) \times \pi(a,u), \sigma(v) \times \pi(a,v))$ (for $\pi_1, \pi_2 \in \{+,-\}$, $\pi_1 \times \pi_2 = +$ if (π_1, π_2) is $(+,+)$ or $(-,-)$ and $\pi_1 \times \pi_2 = -$ if (π_1, π_2) is $(+,-)$ or $(-,+)$).

Step 9 Return $\tilde{G} = (\tilde{V}, \tilde{A}; \tilde{\partial})$.

Remark 6 *A strongly connected component W_i is inconsistent if and only if $i \in I_2$ (see, [3, Corollary 5.4]). The above condensation algorithm can also be regarded as a sequence of something like the condensation for the associated skew-symmetric graph with some operations and transformation from the obtained graph to an associated bidirected graph.*

Let us define the maps $\alpha : V \to \tilde{V}$ and $\beta : \tilde{V} \to V$ by

$$\alpha(v) = \tilde{v}_i \quad (v \in W_i), \qquad \beta(\tilde{v}_i) = w_i \quad (i \in I).$$

Moreover, we define $\alpha(A, \partial) = (A, \tilde{\partial})$ satisfying

$$\tilde{\partial}a = \begin{cases} (\{\alpha(u), \alpha(v)\}, \emptyset) & (\bar{\partial}a = \{u, v\}, \sigma(u) \times \pi(a, u) = +, \sigma(v) \times \pi(a, v) = +), \\ (\{\alpha(u)\}, \{\alpha(v)\}) & (\bar{\partial}a = \{u, v\}, \sigma(u) \times \pi(a, u) = +, \sigma(v) \times \pi(a, v) = -, \\ & \quad \alpha(u) \neq \alpha(v)), \\ (\emptyset, \{\alpha(u), \alpha(v)\}) & (\bar{\partial}a = \{u, v\}, \sigma(u) \times \pi(a, u) = -, \sigma(v) \times \pi(a, v) = -), \\ (\{\alpha(u)\}, \emptyset) & (\bar{\partial}a = \{u\}, \sigma(u) \times \pi(a, u) = +), \\ (\emptyset, \{\alpha(u)\}) & (\bar{\partial}a = \{u\}, \sigma(u) \times \pi(a, u) = -), \end{cases}$$

and $\beta(\tilde{A}, \tilde{\partial}) = (\tilde{A}, \partial)$ satisfying

$$\partial\tilde{a} = \left(\beta(\tilde{\partial}^+\tilde{a}), \beta(\tilde{\partial}^-\tilde{a})\right).$$

Then, we obtain an analogy of [6, Lemma 1] as follows.

Lemma 7 *Let (A', ∂') be a feasible solution of Problem 4 with $F(A', \partial') = \sum_{a' \in A'} |\partial' a'|$ or $F(A', \partial') = |A'|$ on $G = (V, A; \partial)$, then $\alpha(A', \partial')$ is a feasible solution of Problem 4 with the same kind of f on the condensed bidirected graph $\tilde{G}$. Conversely, let $(\tilde{A}', \tilde{\partial}')$ be a feasible solution of Problem 4 with $F(A', \partial') = \sum_{a' \in A'} |\partial' a'|$ or $F(A', \partial') = |A'|$ on $\tilde{G} = (\tilde{V}, \tilde{A}; \tilde{\partial})$, then $\beta(\tilde{A}', \tilde{\partial}')$ is a feasible solution of Problem 4 with the same kind of f on G.*

By virtue of Lemma 7, we can focus on the acyclic case instead of coping with general case owing to the following relation:

$$\mathrm{OPT}(G) = F(A', \partial') \geq F(\alpha(A', \partial')) \geq \mathrm{OPT}(\tilde{G}) = F(\tilde{A}', \tilde{\partial}') = F(\beta(\tilde{A}', \tilde{\partial}')) \geq \mathrm{OPT}(G).$$

4 Minimization on Signs

In this section, we deal with Problem 4 with $F(A', \partial') = \sum_{a' \in A'} |\partial' a'|$.

At first, we give some definitions on bidirected graphs. Let $\gamma(= \gamma(G))$ denote the number of connected components in the underlying graph $\bar{G}$ of G. A vertex $v \in V$ is called a *source* (a *sink*) if v is included in a connected component in $\bar{G}$ which has more than one vertices and any $a \in A$ connected to v in G is positively (negatively) incident to v. The set of sources is denoted by $S(= S(G))$ and that of sinks is denoted by $T(= T(G))$. A vertex $v \in V$ is called an *isolated vertex* if there exists no arcs connected to v. The set of isolated vertices is denoted by $Q(= Q(G))$. A vertex $v \in V$ is called a *pseudo-isolated vertex* if v is the connected component with only one vertex in $\bar{G}$ and there exists a self-loop at v. The set of pseudo-isolated vertices is denoted by $Q'(= Q'(G))$.

When adding an arc $a = (u, v)$ to a bidirected graph G, we write "with proper signs" if signs on a are as follows: $\pi(a, u)$ is equal to $+$ if $\{a \in A \mid u \in \partial^+ a\}$ is empty for the current bidirected graph and $\pi(a, u)$ is equal to $-$ otherwise. The definition of $\pi(a, v)$ is done in the same way.

Now, let us consider Problem 4 with respect to the number of additional signs on an acyclic bidirected graph $G = (V, A; \partial)$. Since a bidirected graph is strongly connected only if its underlying graph is connected, the value of the objective function for a feasible solution must be greater than or equal to $2(\gamma - 1)$. On the other hand, a bidirected graph with $|V| > 1$ is strongly connected only if there are no sources, sinks, isolated vertices and pseudo-isolated vertices. Therefore, the number of additional signs to make a bidirected graph strongly connected is greater than or equal to $|S| + |T| + |Q'| + 2|Q|$. Summing up, we obtain the lower bound $\max\{2(\gamma - 1), |S| + |T| + |Q'| + 2|Q|\}$. Actually, this lower bound can be achieved.

Theorem 8 *Let $G = (V, A; \partial)$ be an acyclic bidirected graph with $|V| > 1$. Then the minimum number of $\sum_{a' \in A'} |\partial' a'|$ such that $G' = (V, A \cup A'; \partial')$ is a strongly connected bidirected graph is $\max\{2(\gamma - 1), |S| + |T| + |Q'| + 2|Q|\}$.*

We now describe an algorithm for constructing an optimal solution (whose size is equal to the lower bound). Let $C_1^1, C_2^1, \ldots, C_{k_1}^1, C_1^2, C_2^2, \ldots, C_{k_2}^2, \ldots, C_1^K, C_2^K, \ldots, C_{k_K}^K$ be the distinct vertex sets of connected components of $\bar{G}$ such that each C_i^j contains just j elements of $S \cup T \cup Q' \cup Q$. Note that $\sum_{i=1}^K k_i = \gamma$ and $\sum_{i=1}^K ik_i = |S| + |T| + |Q'| + |Q|$.

Algorithm ADDITIONAL SIGNS(G)

Step 1 Let $A' := \emptyset$.

Step 2 Let $u_1, u_2, \ldots, u_{L_1}$ ($L_1 := k_1 - |Q|$) be the elements of $\left(\bigcup_{i=1}^{k_1} C_i^1\right) \cap (S \cup T \cup Q')$. If $L_1 = \gamma = 1$, add a self-loop at u_1 with a proper sign and go to Step 6. If $L_1 = \gamma > 1$, add $\{(u_1, u_i) \mid 2 \leq i \leq L_1\}$ to A' with proper signs and go to Step 6.

Step 3 Let $\mathcal{C} = \left\{C_1^2, C_2^2, \ldots, C_{k_2}^2, \ldots, C_1^K, C_2^K, \ldots, C_{k_K}^K\right\}$. For each $C \in \mathcal{C}$, pick up two distinct elements of $C \cap (S \cup T)$ and label them as l_i, r_i ($i = 1, 2, \ldots, |\mathcal{C}|$). Label the rest of the elements of $\bigcup_{C \in \mathcal{C}} C \cap (S \cup T)$ as $w_1, w_2, \ldots, w_{L_2}$ with $L_2 := \sum_{i=3}^K (i-2)k_i$. Add $\{(u_i, w_i) \mid 1 \leq i \leq \min\{L_1, L_2\}\}$ to A' with proper signs.

Step 4 Let $q_1, \ldots, q_{|Q|}$ be the elements of Q and define $l_{|\mathcal{C}|+i} = r_{|\mathcal{C}|+i} = q_i$ for $i = 1, \ldots, |Q|$. Add $\{(r_i, l_{i+1}) \mid 1 \leq i < |\mathcal{C}| + |Q|\}$ to A' with proper signs.

Step 5 Compare L_1 with L_2.

- **Step 5-1** If $L_2 \leq L_1 - 2$, add (u_{L_2+1}, l_1) and $\left\{(u_i, r_{|\mathcal{C}|+|Q|}) \mid L_2 + 1 < i \leq L_1\right\}$ to A' with proper signs.
- **Step 5-2** If $L_2 = L_1 - 1$, add (u_{L_1}, l_1) and a self-loop at $r_{|\mathcal{C}|+|Q|}$ to A' with proper signs.
- **Step 5-3** If $L_2 \geq L_1$, add self-loops at l_1, $r_{|\mathcal{C}|+|Q|}$ and w_i for $i = L_1 + 1, L_1 + 2, \ldots, L_2$ to A' with proper signs.

Step 6 Return $G' = (V, A \cup A'; \partial')$.

It should be noted that $L_2 \leq L_1 - 2$ holds if and only if $2(\gamma - 1) \geq |S| + |T| + |Q'| + 2|Q|$ holds owing to the following relation:

$$L_2 - (L_1 - 2) = \sum_{i=2}^K (i-2)k_i - (k_1 - |Q|) + 2 = \sum_{i=1}^K ik_i - 2\gamma + |Q| + 2 = |S| + |T| + |Q'| + 2|Q| - 2(\gamma - 1).$$

The output of the above algorithm is strongly connected. This can be confirmed by the following facts when $\gamma > L_1$. (It can be shown easier when $\gamma = L_1$.)

- The vertex set $\{l_i, r_i \mid 1 \leq i \leq |\mathcal{C}|\} \cup Q$ is strongly connected.
- Each vertex in $\{l_i, r_i \mid 1 \leq i \leq |\mathcal{C}|\} \cup Q$ is inconsistent.
- For each vertex $v \in V \setminus (\{l_i, r_i \mid 1 \leq i \leq |\mathcal{C}|\} \cup Q)$, there exist $v_1^*, v_2^* \in \{l_i, r_i \mid 1 \leq i \leq |\mathcal{C}|\} \cup Q$, a path P_1 between v and v_1^* and a path P_2 between v and v_2^* such that end arcs of P_1 and P_2 connected to v are oppositely incident.

The above algorithm finds an optimal solution in linear time.

Theorem 9 *Problem 4 with $F(A', \partial') = \sum_{a' \in A'} |\partial' a'|$ can be solved in linear time.*

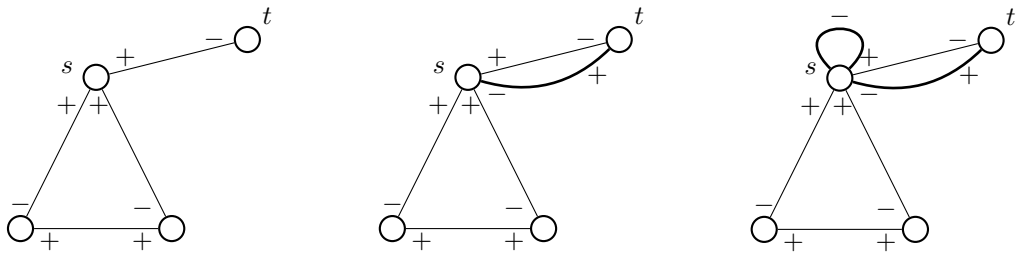
5 Minimization on Arcs

In this section, we deal with Problem 4 with $F(A', \partial') = |A'|$.

Let $\lambda(G)$ be defined by $\lambda(G) := \max\{\gamma - 1, \lceil(|S| + |T| + |Q'|)/2\rceil + |Q|\}$. Clearly, $\lambda(G)$ is the lower bound of Problem 4 with $F(A', \partial') = |A'|$ (which can be derived as well as that for the problem on signs). Unfortunately, however, there is a small example which cannot be made strongly connected by $\lambda(G)$ additional arcs (see Fig. 2), whereas we can always achieve the lower bound when we deal with the number of additional signs as shown in the previous section. For the original graph G in Fig. 2 (a), we have

$$\lambda(G) = \max\left\{\left\lceil\frac{1+1+0}{2}\right\rceil + 0\right\} = \max\{1, 0\} = 1.$$

Since there exist a source s and a sink t in G, if we assume this lower bound can be achieved for G we must add the arc $a = (s, t)$ with proper signs in order to extinguish both source and sink at the same time (see Fig. 2 (b)). However, it is not strongly connected. Actually, the minimum number of additional arcs to make G strongly connected is two and one of the optimal solutions is shown in Fig. 2 (c). On the other hand, there is also an example which can be made strongly connected with $\lambda(G)$ additional arcs.



(a) Original graph. (b) One arc added. (c) Optimal solution.

Figure 2: Example: the size of optimal solution is $\lambda(G) + 1$. Bold lines represent the additional arcs in (b) and (c).

Now we aim at obtaining the upper bound of the size of optimal solutions. Actually, we can show the next theorem.

Theorem 10 *Let $G = (V, A; \partial)$ be an acyclic bidirected graph. Then the minimum number of $|A'|$ such that $G' = (V, A \cup A'; \partial')$ is a strongly connected bidirected graph is $\lambda(G)$ or $\lambda(G) + 1$.*

Note that if the output of ADDITIONAL SIGNS(G) contains at most one self-loop, then it is also an optimal solution of the problem of minimizing the number of additional arcs. If the output of ADDITIONAL SIGNS(G) contains more than 1 self-loops, however, it is not optimal for the problem on arcs. We can construct a feasible solution of size $\lambda(G)$ or $\lambda(G) + 1$ as follows:

Algorithm ADDITIONAL ARCS(G)

Step 1 Let $A' := \emptyset$.

Step 2 Let $u_1, u_2, \ldots, u_{L_1}$ ($L_1 := k_1 - |Q|$) be the elements of $\left(\bigcup_{i=1}^{k_1} C_i^1\right) \cap (S \cup T \cup Q')$. If $L_1 = \gamma = 1$, add a self-loop at u_1 with a proper sign and go to Step 13. If $L_1 = \gamma > 1$, add $\{(u_1, u_i) \mid 2 \leq i \leq L_1\}$ to A' with proper signs and go to Step 13.

Step 3 Let $\mathcal{C} = \left\{C_1^2, C_2^2, \ldots, C_{k_2}^2, \ldots, C_1^K, C_2^K, \ldots, C_{k_K}^K\right\}$. For each $C \in \mathcal{C}$, pick up two distinct elements of $C \cap (S \cup T)$ and label them as l_i, r_i ($i = 1, 2, \ldots, |\mathcal{C}|$). Label the rest of the elements of $\bigcup_{C \in \mathcal{C}} C \cap (S \cup T)$ as $w_1, w_2, \ldots, w_{L_2}$ with $L_2 := \sum_{i=3}^{K}(i-2)k_i$. Add $\{(u_i, w_i) \mid 1 \leq i \leq \min\{L_1, L_2\}\}$ to A' with proper signs.

Step 4 Let $q_1, \ldots, q_{|Q|}$ be the elements of Q and define $l_{|\mathcal{C}|+i} = r_{|\mathcal{C}|+i} = q_i$ for $i = 1, \ldots, |Q|$.

Step 5 If $L_2 \leq L_1 - 2$, add (u_{L_2+1}, l_1), $\{(u_i, r_{|\mathcal{C}|+|Q|}) \mid L_2 + 1 < i \leq L_1\}$ and $\{(r_i, l_{i+1}) \mid 1 \leq i < |\mathcal{C}| + |Q|\}$ to A' with proper signs and go to Step 13.

Step 6 If $L_2 = L_1 - 1$, add (u_{L_1}, l_1), $\{(r_i, l_{i+1}) \mid 1 \leq i < |\mathcal{C}| + |Q|\}$ and a self-loop at $r_{|\mathcal{C}|+|Q|}$ with proper signs and go to Step 13.

Step 7 If $|Q| = 1$, add a new vertex q_2 to V and add (q_1, q_2) with $(+, -)$ to A'. Otherwise, add (q_i, q_{i+1}) to A' with $(+, -)$ for $i = 1, 2, \ldots, |Q| - 1$.

Step 8 Define a new bidirected graph $\hat{G} = (\hat{V}, \hat{A}; \hat{\partial})$ from the bidirected graph $G' = (V, A \cup A'; \partial')$ as follows:

$$\hat{V} := \{l_i, r_i \mid 1 \leq i \leq |\mathcal{C}|\} \cup \{w_j \mid L_1 < j \leq L_2\} \cup \{q_1, q_{\max\{2,|Q|\}}\},$$

$$\hat{A} := \left\{a = (u, v) \text{ with } (\pi_u, \pi_v) \mid \exists(\pi_u, \pi_v)\text{-path from } u \text{ to } v \text{ in } G', \{u, v\} \subseteq \hat{V}\right\}.$$

Step 9 Construct a maximal matching $M = \{m_1, m_2, \ldots, m_{|M|}\}$ ($m_i = (v_i^l, v_i^r)$) in the underlying graph of $\hat{G}$. Add $B := \{(v_i^r, v_{i+1}^l) \mid 1 \leq i \leq |M|$ $(v_{|M|+1}^l := v_1^l)\}$ to A' with proper signs.

Step 10 Let $p_1, p_2, \ldots, p_l$ be the vertices in $\hat{V}$ which are not the endpoints of any element of M. Add $P := \{(p_{2i-1}, p_{2i}) \mid 1 \leq i \leq \lfloor l/2 \rfloor\}$ with proper signs to A'. If l is odd, add a self-loop at p_l to A' with a proper sign.

Step 11 $\tilde{G} \leftarrow$ CONDENSE($G = (V, A \cup A'; \partial')$).

Step 12 Let $\tilde{v}$ be the only one element of $S(\tilde{G}) \cup T(\tilde{G}) \cup Q'(\tilde{G}) \cup Q(\tilde{G})$.

Step 12-1 If $\tilde{v} \in Q'(\tilde{G}) \cup Q(\tilde{G})$, go to Step 13.

Step 12-2 If $\tilde{v} \in S(\tilde{G})$ (resp. $\tilde{v} \in T(\tilde{G})$), add a minus-loop (resp. a plus-loop) at $\beta(\tilde{v})$ to A'.

Step 13 If $|Q| = 1$ holds for the original input graph G, then remove the arc (q_1, q_2) from A'. Return $G' = (V, A \cup A'; \partial')$.

Note that the above algorithm finds a feasible solution of size $\lambda(G)$ or $\lambda(G) + 1$ in linear time.

The cardinality of the solution is $\lambda(G) + 1$ only if Step 12-2 is executed. The above algorithm runs in linear time, thus the total algorithm runs in linear time for a general input bidirected graph.

Theorem 11 *For Problem 4 with $F(A', \partial') = |A'|$, a feasible solution with $|A'| = \lambda(G)$ or $\lambda(G) + 1$ can be found in linear time.*

6 Concluding Remarks

In this paper, we propose two types of problems to make a given bidirected graph strongly connected. The first one aims at minimizing the number of additional signs on arcs and the second one aims at minimizing the number of additional arcs. We give a linear-time algorithm to find an optimal solution for the former problem and a linear-time algorithm to find a feasible solution which can have one more arc than an optimal solution for the latter problem.

As future works, the following problem on minimization on arcs can be considered.

Problem 12 *Let $G = (V, A; \partial)$ be a bidirected graph. Decide whether the minimum number of additional arcs to make G strongly connected is $\lambda(G)$ or $\lambda(G) + 1$.*

Connectivity augmentation problems on bidirected graphs can also be considered, e.g., arc-connectivity augmentation. Let G be k-arc-connected if $G' = (V, A \backslash A^\circ; \partial|(A \backslash A^\circ))$ is strongly connected for all $A^\circ \subseteq A$ with $|A^\circ| = k - 1$.

Problem 13 *Let $G = (V, A; \partial)$ be a bidirected graph and k be a positive integer. Find additional arcs A' and a boundary operator $\partial' : A \cup A' \to 3^V$ $(\partial' a = \partial a \ (\forall a \in A))$ minimizing $F(A', \partial')$ such that G is k-arc-connected.*

Also, the generalization of the problem to make a given undirected graph connected or that to make a given directed graph strongly connected can be considered. Although bidirected graphs can be seen as the common generalization of undirected graphs and directed graphs, the problems in this paper is not the generalization of these classical problems because there is no restriction on additional arcs. For the case of directed graphs, the problem can be formulated as follows:

Problem 14 *Let $G = (V, A; \partial)$ be a bidirected graph. Find additional arcs A' and a boundary operator $\partial' : A \cup A' \to 3^V$ $(\partial' a = \partial a \ (\forall a \in A))$ minimizing $|A'|$ such that $G' := (V, A \cup A'; \partial')$ is a strongly connected bidirected graph and $|\partial'^{+} a'| = |\partial'^{-} a'| = 1$ for all $a' \in A'$.*

References

[1] K. Ando and S. Fujishige, $\sqcup$,$\sqcap$-closed families and signed posets, Discussion Paper Series 567, Institute of Socio-Economic Planning, University of Tsukuba, 1994.

[2] K. Ando and S. Fujishige, On structures of bisubmodular polyhedra, *Mathematical Programming* **74** (1996), 293–317.

[3] K. Ando, S. Fujishige and T. Nemoto, Decomposition of a bidirected graph into strongly connected components and its signed poset structure, *Discrete Applied Mathematics* **68** (1996), 237–248.

[4] M. A. Babenko, Acyclic bidirected and skew-symmetric graphs: algorithms and structure, *International Computer Science Symposium in Russia*, 2006, 23–34.

[5] J. Edmonds and E. L. Johnson, Matching: a well-solved class of linear programs, In: R. Guy, H. Hanani, N. Sauer and J. Schönheim (Eds.): *Combinatorial Structures and Their Applications*, 1970, 88–92.

[6] K. P. Eswaran and R. E. Tarjan, Augmentation problems, *SIAM Journal on Computing* **5** (1976), 653–665.

[7] A. V. Goldberg and A. V. Karzanov, Maximum skew-symmetric flows, In *Proceedings of the Third Annual European Symposium on Algorithms* (1995), 155–170.

[8] A. V. Goldberg and A. V. Karzanov, Path problems in skew-symmetric graphs, *Combinatorica* **16** (1996), 353–382.

[9] A. V. Goldberg and A. V. Karzanov, Maximum skew-symmetric flows and matchings, *Mathematical Programming* **100** (2004), 537–568.

[10] F. Harary, On the notion of balance of a signed graph, *Michigan Mathematical Journal* **2** (1955), 143–146.

[11] S. Iwata, Block triangularization of skew-symmetric matrices, *Linear Algebra and its Applications* **273** (1998), 215–226.

[12] R. M. Karp, Reducibility among combinatorial problems, In: R. E. Miller, J. W. Thatcher (Eds.): *Complexity of Computer Computations*, 1972, 85–103.

[13] W. T. Tutte, Antisymmetrical digraphs, *Canadian Journal of Mathematics* **19** (1967), 1101–1117.

[14] T. Zaslavsky, Orientation of signed graphs, *European Journal of Combinatorics* **12** (1991), 361–375.

Multiple Exchange in $M^\natural$-concave Functions and Its Implication in Economics

Kazuo Murota[1]

School of Business Administration
Tokyo Metropolitan University
Minami-osawa, Hachioji, Tokyo 192-0397, Japan
murota@tmu.ac.jp

Abstract: The multiple exchange property for matroid bases is generalized for $M^\natural$-concave set functions and valuated matroids. The proof is based on the Fenchel-type duality theorem in discrete convex analysis. The present result has an implication in economics: The strong no complementarities (SNC) condition of Gul and Stacchetti is in fact equivalent to the gross substitutes (GS) condition of Kelso and Crawford.

Keywords: matroid, exchange property, discrete convex analysis, gross substitutes condition

1 Introduction

Discrete convex analysis [5, 16, 17] offers a general framework of discrete optimization, combining the ideas from submodular/matroid theory and convex analysis. It has found applications in many different areas including mathematical economics and game theory [6, 10, 12, 18]. The interaction between discrete convex analysis and mathematical economics was initiated by [2] (see also [16, Chapter 11]) and accelerated by the crucial observation of Fujishige–Yang [7] that $M^\natural$-concavity is equivalent to the gross substitutability (GS) of Kelso–Crawford [11].

In matroid theory, one of the classical results [1, 8, 22] says that the basis family of a matroid enjoys the multiple exchange property: For two bases X and Y in a matroid and a subset $I \subseteq X \setminus Y$, there exists a subset $J \subseteq Y \setminus X$ such that $(X \setminus I) \cup J$ and $(Y \setminus J) \cup I$ are both bases. As a quantitative version of this, we may naturally consider the multiple exchange property for a set function f: For two subsets X, Y and a subset $I \subseteq X \setminus Y$, there exists $J \subseteq Y \setminus X$ such that $f(X) + f(Y) \leq f((X \setminus I) \cup J) + f((Y \setminus J) \cup I)$.

The objective of this paper is to establish this multiple exchange property for $M^\natural$-concave set functions. The results are described in Section 2 and the proof, based on the Fenchel-type duality theorem in discrete convex analysis, is given in Section 3. Our result settles an old question in economics: The strong no complementarities (SNC) condition of Gul and Stacchetti [9] is in fact equivalent to the gross substitutes condition, which is discussed in Section 4.

2 Results

Let N be a finite set, say, $N = \{1, 2, \ldots, n\}$. For a function $f : 2^N \to \mathbb{R} \cup \{-\infty\}$, $\operatorname{dom} f$ denotes the effective domain of f, i.e., $\operatorname{dom} f = \{X \mid f(X) > -\infty\}$.

A function $f : 2^N \to \mathbb{R} \cup \{-\infty\}$ is called *$M^\natural$-concave* [16, 19], if, for any $X, Y \in \operatorname{dom} f$ and $i \in X \setminus Y$, we have (i) $X - i \in \operatorname{dom} f$, $Y + i \in \operatorname{dom} f$ and

$$f(X) + f(Y) \leq f(X - i) + f(Y + i), \tag{1}$$

[1]This work was supported by The Mitsubishi Foundation, CREST, JST, and JSPS KAKENHI Grant Number 26280004.

or (ii) there exists some $j \in Y \setminus X$ such that $X - i + j \in \operatorname{dom} f$, $Y + i - j \in \operatorname{dom} f$ and

$$f(X) + f(Y) \leq f(X - i + j) + f(Y + i - j). \tag{2}$$

Here we use short-hand notations $X - i = X \setminus \{i\}$, $Y + i = Y \cup \{i\}$, $X - i + j = (X \setminus \{i\}) \cup \{j\}$, and $Y + i - j = (Y \cup \{i\}) \setminus \{j\}$. This property is referred to as the *exchange property*. The exchange property can be expressed more compactly as:

(M$^\natural$-EXC) [Exchange property] For any $X, Y \subseteq N$ and $i \in X \setminus Y$, we have

$$f(X) + f(Y) \leq \max\Big[f(X - i) + f(Y + i), \max_{j \in Y \setminus X}\{f(X - i + j) + f(Y + i - j)\}\Big], \tag{3}$$

where $(-\infty) + a = a + (-\infty) = (-\infty) + (-\infty) = -\infty$ for $a \in \mathbb{R}$, $-\infty \leq -\infty$, and a maximum taken over the empty set is defined to be $-\infty$.

In this paper we are concerned with the *multiple exchange property*:

(M$^\natural$-EXC$_{\mathrm{m}}$) [Multiple exchange property] For any $X, Y \subseteq N$ and $I \subseteq X \setminus Y$, there exists $J \subseteq Y \setminus X$ such that $f(X) + f(Y) \leq f((X \setminus I) \cup J) + f((Y \setminus J) \cup I)$, i.e.,

$$f(X) + f(Y) \leq \max_{J \subseteq Y \setminus X}\{f((X \setminus I) \cup J) + f((Y \setminus J) \cup I)\}. \tag{4}$$

Theorem 1 *Every M$^\natural$-concave function $f : 2^N \to \mathbb{R} \cup \{-\infty\}$ has the multiple exchange property* (M$^\natural$-EXC$_{\mathrm{m}}$).

Proof: The proof is given in Section 3. □

The concept of valuated matroid due to Dress–Wenzel [3, 4] (see also [15, Chapter 5]) is defined in terms of an exchange property similar to (M$^\natural$-EXC). A function $f : 2^N \to \mathbb{R} \cup \{-\infty\}$ is called a *valuated matroid*, if, for any $X, Y \subseteq N$ and $i \in X \setminus Y$, it holds that

$$f(X) + f(Y) \leq \max_{j \in Y \setminus X}\{f(X - i + j) + f(Y + i - j)\}. \tag{5}$$

A valuated matroid is nothing but an M$^\natural$-concave function f such that $\operatorname{dom} f$ consists of equi-cardinal subsets, i.e., $|X| = |Y|$ for any $X, Y \in \operatorname{dom} f$. In this case, $\operatorname{dom} f$ forms the basis family of a matroid on N. As a corollary of Theorem 1 we obtain the following.

Theorem 2 *Every valuated matroid f has the multiple exchange property* (M$^\natural$-EXC$_{\mathrm{m}}$) *with $|J| = |I|$.*

This theorem contains, as a special case, the multiple exchange theorem for matroid bases due to Brylawski [1], Greene [8] and Woodall [22]; see also [13, 14, 20].

Theorem 3 ([1, 8, 22]) *Let X and Y be bases in a matroid, and let $I \subseteq X \setminus Y$. Then there exists a subset $J \subseteq Y \setminus X$ such that $(X \setminus I) \cup J$ and $(Y \setminus J) \cup I$ are both bases.*

The converse of Theorem 1 is also true and thus we obtain a characterization of M$^\natural$-concave functions (Theorem 4 below) in terms of the multiple exchange property. It is noted that the converse of Theorem 1 seems intuitively obvious, but a formal proof (which is omitted here) is needed, since we have to assure that for $I = \{i\}$ in (M$^\natural$-EXC$_{\mathrm{m}}$) there exists J with $|J| \leq 1$.

Theorem 4 *A function $f : 2^N \to \mathbb{R} \cup \{-\infty\}$ is M$^\natural$-concave if and only if it has the multiple exchange property* (M$^\natural$-EXC$_{\mathrm{m}}$).

3 Proof of Theorem 1

In this section we give a proof to the main theorem, Theorem 1. Let $f : 2^N \to \mathbb{R} \cup \{-\infty\}$ be an $M^\natural$-concave function, $X, Y \in \operatorname{dom} f$ and $I \subseteq X \setminus Y$. We shall prove

$$f(X) + f(Y) \leq \max_{J \subseteq Y \setminus X} \{f((X \setminus I) \cup J) + f((Y \setminus J) \cup I)\}. \tag{6}$$

Our proof is based on the Fenchel-type duality theorem in discrete convex analysis.

With the notations

$$\begin{aligned} &C = X \cap Y, \qquad X_0 = X \setminus Y = X \setminus C, \qquad Y_0 = Y \setminus X = Y \setminus C, \\ &f_1(J) = f((X \setminus I) \cup J) = f((X_0 \setminus I) \cup C \cup J) \qquad (J \subseteq Y_0), \\ &f_2(J) = f((Y \setminus J) \cup I) = f(I \cup C \cup (Y_0 \setminus J)) \qquad (J \subseteq Y_0), \end{aligned}$$

the inequality (6) is rewritten as

$$f(X) + f(Y) \leq \max_{J \subseteq Y_0} \{f_1(J) + f_2(J)\}. \tag{7}$$

Both f_1 and f_2 are $M^\natural$-concave set functions on Y_0.

Consider the (convex) conjugate functions of f_1 and f_2 given by

$$\begin{aligned} g_1(q) &= \max_{J \subseteq Y_0} \{f_1(J) - q(J)\} \qquad (q \in \mathbb{R}^{Y_0}), \\ g_2(q) &= \max_{J \subseteq Y_0} \{f_2(J) - q(J)\} \qquad (q \in \mathbb{R}^{Y_0}), \end{aligned}$$

where $q(J) = \sum_{i \in J} q_j$. For any $J \subseteq Y_0$ and $q \in \mathbb{R}^{Y_0}$, we have

$$\begin{aligned} f_1(J) + f_2(J) &= (f_1(J) - q(J)) + (f_2(J) + q(J)) \\ &\leq \max_{J \subseteq Y_0} \{f_1(J) - q(J)\} + \max_{J \subseteq Y_0} \{f_2(J) + q(J)\} \\ &= g_1(q) + g_2(-q). \end{aligned}$$

The Fenchel-type duality theorem in discrete convex analysis [16, Theorem 8.21 (1)] asserts that there exist J and q for which the above inequality holds in equality, i.e.,

$$\max_{J \subseteq Y_0} \{f_1(J) + f_2(J)\} = \min_{q \in \mathbb{R}^{Y_0}} \{g_1(q) + g_2(-q)\}. \tag{8}$$

Note that $\operatorname{dom} g_1 = \operatorname{dom} g_2 = \mathbb{R}^{Y_0}$ and the assumption in [16, Theorem 8.21 (1)] is satisfied.

The desired inequality (7) follows from (8) and Lemma 5 below.

Lemma 5 *For any $q \in \mathbb{R}^{Y_0}$, we have $g_1(q) + g_2(-q) \geq f(X) + f(Y)$.*

Proof: Let g be the (convex) conjugate function of f, i.e.,

$$g(p) = \max_{Z \subseteq N} \{f(Z) - p(Z)\} \qquad (p \in \mathbb{R}^N).$$

By the conjugacy theorem in discrete convex analysis ([17, Theorem 3.4], [16, Theorems 8.4, (8.10)]), g is a polyhedral $L^\natural$-convex function. In particular, it is submodular:

$$g(p) + g(p') \geq g(p \vee p') + g(p \wedge p') \qquad (p, p' \in \mathbb{R}^N), \tag{9}$$

where $p \vee p'$ and $p \wedge p'$ denote the vectors of component-wise maximum and minimum, i.e.,

$$(p \vee p')_i = \max(p_i, p'_i), \qquad (p \wedge p')_i = \min(p_i, p'_i).$$

For a vector $q \in \mathbb{R}^{Y_0}$ we define $p^{(1)}, p^{(2)} \in \mathbb{R}^N$ by

$$p_i^{(1)} = \begin{cases} q_i & (i \in Y_0), \\ -M & (i \in X_0 \setminus I), \\ +M & (i \in I), \\ -M & (i \in C), \\ +M & (i \in N \setminus (X \cup Y)), \end{cases} \qquad p_i^{(2)} = \begin{cases} q_i & (i \in Y_0), \\ +M & (i \in X_0 \setminus I), \\ -M & (i \in I), \\ -M & (i \in C), \\ +M & (i \in N \setminus (X \cup Y)), \end{cases}$$

where M is a sufficiently large positive number. Then we have

$$\begin{aligned} g_1(q) &= \max_{J \subseteq Y_0} \{f((X_0 \setminus I) \cup C \cup J) - q(J)\} \\ &= g(p^{(1)}) - M(|X_0 \setminus I| + |C|), \\ g_2(-q) &= \max_{J \subseteq Y_0} \{f(I \cup C \cup (Y_0 \setminus J)) + q(J)\} \\ &= \max_{K \subseteq Y_0} \{f(I \cup C \cup K) - q(K)\} + q(Y_0) \\ &= g(p^{(2)}) - M(|I| + |C|) + q(Y_0). \end{aligned}$$

By adding these two and using submodularity (9) of g, we obtain

$$\begin{aligned} g_1(q) + g_2(-q) &= g(p^{(1)}) + g(p^{(2)}) - M(|X| + |C|) + q(Y_0) \\ &\geq g(p^{(1)} \vee p^{(2)}) + g(p^{(1)} \wedge p^{(2)}) - M(|X| + |C|) + q(Y_0). \end{aligned} \tag{10}$$

Since

$$(p^{(1)} \vee p^{(2)})_i = \begin{cases} q_i & (i \in Y_0), \\ +M & (i \in X_0 \setminus I), \\ +M & (i \in I), \\ -M & (i \in C), \\ +M & (i \in N \setminus (X \cup Y)), \end{cases} \qquad (p^{(1)} \wedge p^{(2)})_i = \begin{cases} q_i & (i \in Y_0), \\ -M & (i \in X_0 \setminus I), \\ -M & (i \in I), \\ -M & (i \in C), \\ +M & (i \in N \setminus (X \cup Y)), \end{cases}$$

we have

$$g(p^{(1)} \vee p^{(2)}) \geq f(Y) - q(Y_0) + M|C|, \tag{11}$$

$$g(p^{(1)} \wedge p^{(2)}) \geq f(X) + M|X|. \tag{12}$$

The substitution of (11) and (12) into (10) yields the inequality $g_1(q) + g_2(-q) \geq f(X) + f(Y)$. □

Remark 6 Among several different proofs known for the multiple exchange property of matroid bases (Theorem 3), the proofs of Woodall [22] and McDiarmid [14] are based on minimax duality formulas for matroid rank functions (matroid union/intersection theorems). Our proof of Theorem 1 generalizes this idea to $\mathrm{M}^\natural$-concave functions. Note that matroid rank functions are $\mathrm{M}^\natural$-concave functions [17], and the matroid union/intersection theorems are special cases of the Fenchel-type duality theorem for $\mathrm{M}^\natural$-concave functions [16, Section 8.2.3].

Remark 7 The above proof shows that the subset J in ($\mathrm{M}^\natural$-EXC_m) can be computed in polynomial time by an adaptation of the valuated matroid intersection algorithm [15, Chapter 5].

4 An Implication in Economics

For a vector $p \in \mathbb{R}^N$ we define

$$D(p|f) = \arg\max_X \{f(X) - p(X) \mid X \subseteq N\}, \tag{13}$$

where $p(X) = \sum_{i \in X} p_i$. In economic applications where f denotes a utility (valuation) function over indivisible goods, p is interpreted as the vector of prices and $D(p) = D(p|f)$ represents the demand correspondence.

Kelso and Crawford [11] introduced the following property for $f : 2^N \to \mathbb{R} \cup \{-\infty\}$, which turned out to be the key property in discussing economies with indivisible goods[1]:

> **(GS)** [Gross Substitutes property] For any vectors p and q with $p \leq q$ and $X \in D(p|f)$, there exists $Y \in D(q|f)$ such that $\{i \in X \mid p_i = q_i\} \subseteq Y$.

Gul and Stacchetti [9] considered the following three properties:

> **(SI)** [Single Improvement property] For any $p \in \mathbb{R}^N$, if $X \notin D(p|f)$, there exists $Y \subseteq N$ such that $|X \setminus Y| \leq 1$, $|Y \setminus X| \leq 1$, and $f[-p](X) < f[-p](Y)$,
>
> **(NC)** [No Complementarities property] For any $p \in \mathbb{R}^N$, if $X, Y \in D(p|f)$ and $I \subseteq X \setminus Y$, there exists $J \subseteq Y \setminus X$ such that $(X \setminus I) \cup J \in D(p|f)$,
>
> **(SNC)** [Strong No Complementarities property] For $X, Y \subseteq N$ and $I \subseteq X \setminus Y$, there exists $J \subseteq Y \setminus X$ such that $f(X) + f(Y) \leq f((X \setminus I) \cup J) + f((Y \setminus J) \cup I)$.

They showed that (NC) and (SI) are equivalent to (GS), and these (mutually equivalent) conditions are implied by (SNC). Subsequently, Fujishige and Yang [7] pointed out that (GS) is equivalent to ($\mathrm{M}^\natural$-EXC) for $\mathrm{M}^\natural$-concavity. These results are summarized schematically here as:

$$(\mathrm{SNC}) \Longrightarrow (\mathrm{NC}) \iff (\mathrm{GS}) \iff (\mathrm{SI}) \iff (\mathrm{M}^\natural\text{-EXC}). \tag{14}$$

Since (SNC) and ($\mathrm{M}^\natural$-EXC_m) are mathematically the same, and ($\mathrm{M}^\natural$-EXC_m) follows from ($\mathrm{M}^\natural$-EXC) by Theorem 1, we now see that the above five properties are in fact equivalent:

$$(\mathrm{SNC}) \iff (\mathrm{NC}) \iff (\mathrm{GS}) \iff (\mathrm{SI}) \iff (\mathrm{M}^\natural\text{-EXC}). \tag{15}$$

In this context it would be natural to consider the following simultaneous version of (NC):

> **(NCsim)** For any $p \in \mathbb{R}^N$, if $X, Y \in D(p|f)$ and $I \subseteq X \setminus Y$, there exists $J \subseteq Y \setminus X$ such that $(X \setminus I) \cup J \in D(p|f)$ and $(Y \setminus J) \cup I \in D(p|f)$.

Obviously, (SNC) $\Longrightarrow$ (NCsim) and (NCsim) $\Longrightarrow$ (NC). Hence (NCsim) is also equivalent to (GS).

We conclude this section by stating the equivalence of all the six properties as a theorem.

Theorem 8 *For a function $f : 2^N \to \mathbb{R} \cup \{-\infty\}$, we have the following equivalence:*

$$(\mathrm{M}^\natural\text{-}\mathrm{EXC}_\mathrm{m}) = (\mathrm{SNC}) \iff (\mathrm{NCsim}) \iff (\mathrm{NC}) \iff (\mathrm{GS}) \iff (\mathrm{SI}) \iff (\mathrm{M}^\natural\text{-EXC}).$$

Acknowledgements: The author thanks Akiyoshi Shioura, Akihisa Tamura and Yu Yokoi for discussion and comments.

References

[1] T. H. Brylawski, Some properties of basic families of subsets, *Discrete Mathematics* **6** (1973), 333–341.

[2] V. Danilov, G. Koshevoy, and K. Murota, Discrete convexity and equilibria in economies with indivisible goods and money, *Mathematical Social Sciences* **41** (2001), 251–273.

[1] To be precise, Kelso and Crawford [11] as well as Gul and Stacchetti [9] and Fujishige and Yang [7] treat the case of $f : 2^N \to \mathbb{R}$, with $\mathrm{dom}\, f = 2^N$. It can be verified that (14) is true for $f : 2^N \to \mathbb{R} \cup \{-\infty\}$.

[3] A. W. M. Dress and W. Wenzel, Valuated matroid: A new look at the greedy algorithm, *Applied Mathematics Letters* **3** (1990), 33–35.

[4] A. W. M. Dress and W. Wenzel, Valuated matroids, *Advances in Mathematics* **93** (1992), 214–250.

[5] S. Fujishige, *Submodular Functions and Optimization*, 2nd ed., Elsevier, Amsterdam, 2005.

[6] S. Fujishige and A. Tamura, A two-sided discrete-concave market with possibly bounded side payments: An approach by discrete convex analysis, *Mathematics of Operations Research* **32** (2007), 136–155.

[7] S. Fujishige and Z. Yang, A note on Kelso and Crawford's gross substitutes condition, *Mathematics of Operations Research* **28** (2003), 463–469.

[8] C. Greene, A multiple exchange property for bases, *Proc. American Mathematical Society* **39** (1973), 45–50.

[9] F. Gul and E. Stacchetti, Walrasian equilibrium with gross substitutes, *Journal of Economic Theory* **87** (1999), 95–124.

[10] Y. T. Ikebe, Y. Sekiguchi, A. Shioura and, A. Tamura, Stability and competitive equilibria in multi-unit trading networks with discrete concave utility functions, *Japan Journal of Industrial and Applied Mathematics* **32** (2015), 373–410.

[11] A. S. Kelso, Jr., and V. P. Crawford, Job matching, coalition formation, and gross substitutes, *Econometrica* **50** (1982), 1483–1504.

[12] F. Kojima, A. Tamura and M. Yokoo, Designing matching mechanisms under constraints: An approach from discrete convex analysis. The 7th International Symposium on Algorithmic Game Theory, Patras, 2014; https://mpra.ub.uni-muenchen.de/56189/

[13] J. P. S. Kung, Basis-exchange properties, in: N. White, ed., *Theory of Matroids*, Cambridge University Press, London, 1986, Chapter 4, 62–75.

[14] C. J. H. McDiarmid, An exchange theorem for independence structures, *Proc. American Mathematical Society* **47** (1975), 513–514.

[15] K. Murota, *Matrices and Matroids for Systems Analysis*, Springer, Berlin, 2000.

[16] K. Murota, *Discrete Convex Analysis*, SIAM, Philadelphia, 2003.

[17] K. Murota, Recent developments in discrete convex analysis, in: W. J. Cook, L. Lovász, and J. Vygen (eds.), *Research Trends in Combinatorial Optimization*, Springer, Berlin, 2009, pp. 219–260.

[18] K. Murota, Discrete convex analysis: A tool for economics and game theory, Journal of Mechanism and Institution Design, **1** (2016), 151-273.

[19] K. Murota and A. Shioura, M-convex function on generalized polymatroid, *Mathematics of Operations Research* **24** (1999), 95–105.

[20] A. Schrijver, *Combinatorial Optimization—Polyhedra and Efficiency*, Springer, Heidelberg, 2003.

[21] É. Tardos, Generalized matroids and supermodular colourings, in: A. Recski and L. Lovász (eds.), *Matroid Theory*, Colloquia Mathematica Societatis János Bolyai, **40**, North-Holland, Amsterdam, 1985, 359–382.

[22] D. R. Woodall, An exchange theorem for bases of matroids, *Journal of Combinatorial Theory* **B16** (1974), 227–228.

Time Bounds of Two-Phase Algorithms for L-convex Function Minimization[1]

Kazuo Murota

School of Business Administration,
Tokyo Metropolitan University,
Tokyo 192-0397, Japan
murota@tmu.ac.jp

Akiyoshi Shioura

Department of Industrial Engineering
and Economics,
Tokyo Institute of Technology,
Tokyo 152-8550, Japan
shioura.a.aa@m.titech.ac.jp

Abstract: We analyze minimization algorithms, called the two-phase algorithms, for $\mathrm{L}^\natural$-convex functions in discrete convex analysis and derive tight bounds for the number of iterations.

Keywords: discrete optimization, iterative auction, discrete convex function

1 Introduction and Result

With motivations from auction theory, we discuss minimization of a discrete convex function called $\mathrm{L}^\natural$-convex function. A function $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ defined on integer lattice points is said to be *$L^\natural$-convex* [10] if for every $p, q \in \operatorname{dom} g$ and every nonnegative $\lambda \in \mathbb{Z}_+$, it holds that

$$g(p) + g(q) \geq g((p + \lambda\mathbf{1}) \wedge q) + g(p \vee (q - \lambda\mathbf{1})), \tag{1}$$

where $\operatorname{dom} g = \{p \in \mathbb{Z}^n \mid g(p) < +\infty\}$, $\mathbf{1} = (1, 1, \ldots, 1)$, and for $p, q \in \mathbb{Z}^n$ the vectors $p \wedge q$ and $p \vee q$ denote, respectively, the vectors of component-wise minimum and maximum of p and q. The concept of $\mathrm{L}^\natural$-convex function plays a primary role in the theory of discrete convex analysis [10], and an important application can be found in auction theory, in addition to discrete optimization and computer vision (see [11, 15]).

We consider a certain type of algorithm for $\mathrm{L}^\natural$-convex function minimization, called the two-phase algorithm. While the two-phase algorithm and its variants are originally considered in [13, 14] for a specific $\mathrm{L}^\natural$-convex function arising from an auction model (see Section 2 for details), the algorithms work for general $\mathrm{L}^\natural$-convex functions.

As its name indicates, the two-phase algorithm consists of two phases, the up phase and the down phase. The algorithm starts from an arbitrarily chosen initial vector, and the vector moves upward in the up phase and then downward in the down phase. A detailed description of the algorithm is as follows, where $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ is an $\mathrm{L}^\natural$-convex function, $N = \{1, 2, \ldots, n\}$, and $\chi_X \in \{0, 1\}^n$ is the characteristic vector of a set $X \subseteq N$.

Algorithm TwoPhase

Step 0: Find a vector $p^\circ \in \operatorname{dom} g$ and set $p := p^\circ$. Go to Up Phase.

Up Phase:

Step U1: Find a minimizer $X \subseteq N$ of $g(p + \chi_X)$.

Step U2: If $g(p + \chi_X) = g(p)$, then go to Down Phase.

Step U3: Set $p := p + \chi_X$, and go to Step U1.

[1]This work was supported by The Mitsubishi Foundation, CREST, JST, Grant Number JPMJCR14D2, and JSPS KAKENHI Grant Numbers 26280004, 15K00030, 15H00848.

Down Phase:
Step D1: Find a minimizer $X \subseteq N$ of $g(p - \chi_X)$.
Step D2: If $g(p - \chi_X) = g(p)$, then output p and stop.
Step D3: Set $p := p - \chi_X$, and go to Step D1.

Note that Steps U1 and D1 in each iteration can be done in polynomial time (in n) by reduction to the minimization of submodular set functions ρ_p^+, ρ_p^- given by

$$\rho_p^+(X) = g(p + \chi_X), \qquad \rho_p^-(X) = g(p - \chi_X) \qquad (X \subseteq N);$$

see, e.g., [10] for the strongly-polynomial time algorithms for submodular set function minimization.

In this paper, we analyze the number of updates of vector p in the two-phase algorithm[1]. We denote by $\arg\min g$ the set of minimizers of g, and assume $\arg\min g \neq \emptyset$ throughout this paper. We denote

$$\mu(p^\circ) = \min\{\eta(p^*, p^\circ) \mid p^* \in \arg\min g\},$$

where

$$\begin{aligned} \eta(p,q) &= \|p-q\|_\infty^+ + \|p-q\|_\infty^- && (p, q \in \mathbb{Z}^n), \\ \|q\|_\infty^+ &= \max(0, q(1), q(2), \ldots, q(n)) && (q \in \mathbb{Z}^n), \\ \|q\|_\infty^- &= \max(0, -q(1), -q(2), \ldots, -q(n)) && (q \in \mathbb{Z}^n). \end{aligned} \tag{2}$$

That is, $\mu(p^\circ)$ is the η-distance from vector p° to the nearest minimizer of g.

Theorem 1. *Suppose that the algorithm* TwoPhase *is applied to an $L^\natural$-convex function g with an initial vector $p^\circ \in \operatorname{dom} g$. Then, the number of updates of vector p is bounded by $\mu(p^\circ)$ in the up phase and by $\mu(p^\circ)$ in the down phase; in total, bounded by $2\mu(p^\circ)$.*

The proof of Theorem 1 is given in Section 3.1. It is noted that while the minimizers X selected in Steps U1 and D1 are not uniquely determined, the bounds in Theorem 1 hold irrespective of the choice of X. It can be shown by a numerical example that the bounds in the theorem above are tight.

We consider a variant of TwoPhase, named TwoPhaseMinMax in [14]. The algorithm TwoPhaseMinMax is different from TwoPhase in the choice of X in each iteration and the termination condition of each phase. It is obtained from TwoPhase by changing Steps U1, U2, D1, and D2 to the following:

Step U1: Find the unique minimal minimizer $X \subseteq N$ of $g(p + \chi_X)$.
Step U2: If $X = \emptyset$, then go to Down Phase.
Step D1: Find the unique minimal minimizer $X \subseteq N$ of $g(p - \chi_X)$.
Step D2: If $X = \emptyset$, then output p and stop.

Note that in the up phase of TwoPhaseMinMax, the termination condition $X = \emptyset$ can be replaced by the condition $g(p + \chi_X) = g(p)$ since X is the unique minimal minimizer of $g(p + \chi_X)$. Similarly, the termination condition $X = \emptyset$ of the down phase in TwoPhaseMinMax can be replaced by the condition $g(p - \chi_X) = g(p)$. This shows that TwoPhaseMinMax can be regarded as a special implementation of TwoPhase, and hence the following bounds can be obtained from Theorem 1.

Corollary 2. *Suppose that the algorithm* TwoPhaseMinMax *is applied to an $L^\natural$-convex function g with an initial vector $p^\circ \in \operatorname{dom} g$. Then, the number of updates of vector p is bounded by $\mu(p^\circ)$ in the up phase and by $\mu(p^\circ)$ in the down phase; in total, bounded by $2\mu(p^\circ)$.*

We also consider another variant of TwoPhase, named TwoPhaseMinMin in [14], for finding the (unique) minimal minimizer of an $L^\natural$-convex function[2]. We assume the existence of the minimal minimizer in discussing this algorithm. The up phase of the algorithm TwoPhaseMinMin is the same as that of TwoPhaseMinMax, while the down phase of TwoPhaseMinMin is obtained from that of TwoPhaseMinMax by changing Step D1 to the following:

[1] A weaker statement than Theorem 1 is given in an unpublished technical report [13].
[2] Due to $L^\natural$-convexity, a minimal minimizer is uniquely determined if it exists.

Step D1: Find the unique *maximal* minimizer $X \subseteq N$ of $g(p - \chi_X)$.

The following bounds for the numbers of updates of p are shown[3] in [14], where $p^*_{\min} \in \mathbb{Z}^n$ denotes the unique minimal minimizer of $\mathrm{L}^\natural$-convex function g.

Proposition 3 ([14, Theorem 4.13])**.** *Suppose that the algorithm* TwoPhaseMinMin *is applied to an $L^\natural$-convex function g with an initial vector $p^\circ \in \operatorname{dom} g$. Then, the number of updates of vector p is bounded by $\eta(p^\circ,\, p^*_{\min})$ in the up phase and by $2\eta(p^\circ,\, p^*_{\min})$ in the down phase; in total, bounded by $3\eta(p^\circ,\, p^*_{\min})$.*

By Theorem 1, we can improve the bound on the number of updates in the down phase. The behavior of TwoPhaseMinMin applied to an $\mathrm{L}^\natural$-convex function g is the same as that of TwoPhase applied to the $\mathrm{L}^\natural$-convex function $g_\varepsilon(p) = g(p) + \varepsilon \sum_{i=1}^n p(i)$ with a sufficiently small positive ε. Indeed, we have the following equivalences:

$$\begin{aligned} X \subseteq N \text{ is a minimizer of } g_\varepsilon(p + \chi_X) &\iff X \text{ is the minimal minimizer of } g(p + \chi_X), \\ X \subseteq N \text{ is a minimizer of } g_\varepsilon(p - \chi_X) &\iff X \text{ is the maximal minimizer of } g(p - \chi_X), \\ p \in \mathbb{Z}^n \text{ is a minimizer of } g_\varepsilon &\iff p \text{ is the minimal minimizer of } g. \end{aligned}$$

These facts, together with Theorem 1, imply the following bounds.

Proposition 4. *Suppose that the algorithm* TwoPhaseMinMin *is applied to an $L^\natural$-convex function g with an initial vector $p^\circ \in \operatorname{dom} g$. Then, the number of updates of vector p is bounded by $\eta(p^\circ,\, p^*_{\min})$ in the up phase and by $\eta(p^\circ,\, p^*_{\min})$ in the down phase; in total, bounded by $2\eta(p^\circ,\, p^*_{\min})$.*

By a different approach without using perturbation, we can further improve the bound for the down phase.

Theorem 5. *Suppose that the algorithm* TwoPhaseMinMin *is applied to an $L^\natural$-convex function g with an initial vector $p^\circ \in \operatorname{dom} g$. Then, the number of updates of vector p is bounded by $\mu(p^\circ)$ in the up phase and by $\|p^\circ - p^*_{\min}\|^+_\infty$ in the down phase; in total, bounded by $\mu(p^\circ) + \|p^\circ - p^*_{\min}\|^+_\infty$.*

The proof of Theorem 5 is given in Section 3.2. We note that for every $p^\circ \in \mathbb{Z}^n$ it holds that

$$\mu(p^\circ) \le \eta(p^\circ,\, p^*_{\min}), \qquad \|p^\circ - p^*_{\min}\|^+_\infty \le \eta(p^\circ,\, p^*_{\min}).$$

Hence, the bounds in Theorem 5 are indeed better than those in Proposition 4.

2 Motivation from Auction Theory

This research is motivated by design and analysis of iterative auction in auction theory. In the auction literature an algorithm (a mechanism, more precisely) called iterative auction (also called dynamic auction, Walrasian tâtonnement process, etc.) is often used to find equilibrium prices of goods (see, e.g., [3, 4]). An iterative auction updates prices repeatedly by using bidders' demand information, and finds equilibrium prices. A well-known iterative auction is the English auction for a single item.

Let us consider an auction market with n types of items, denoted by $N = \{1, 2, \ldots, n\}$, and m bidders, denoted by $M = \{1, 2, \ldots, m\}$. Each bidder $i \in M$ has his/her valuation function $f_i : 2^N \to \mathbb{Z}$ with the value $f_i(X)$ representing the degree of satisfaction for an item set $X \subseteq N$. We assume that each f_i is an integer-valued nondecreasing function satisfying the so-called "gross-substitutes" condition, which is a natural assumption for valuation functions (see [2, 6, 7] for the precise definition). An allocation of items is defined as a family of item sets $X_1, X_2, \ldots, X_m$ satisfying $X_i \cap X_h = \emptyset$ if $i \neq h$ and $\bigcup_{i \in M} X_i \subseteq N$.

[3]While the algorithm TwoPhaseMinMin in [14] is proposed as an algorithm for a specific $\mathrm{L}^\natural$-convex function (i.e., Lyapunov function), the algorithm as well as its analysis can be naturally extended to general $\mathrm{L}^\natural$-convex functions.

The goal of an auction is to find equilibrium allocation and prices of items. A pair of price vector $p^* \in \mathbb{Z}^n_+$ and an allocation of items $X^*_1, X^*_2, \ldots, X^*_m$ is called a *Walrasian equilibrium* [3, 4] if the following conditions hold:

$$X^*_i \in \arg\max\{f_i(X) - \sum_{j \in X} p^*(j) \mid X \subseteq N\} \qquad (i \in M),$$
$$p^*(j) = 0 \qquad (j \in N \setminus \bigcup_{i \in M} X^*_i).$$

Hence, in the equilibrium each bidder gets his/her best item set and all unsold items have zero price.

The natural and popular iterative auctions are ascending and descending auctions, in which prices are monotonically increasing or decreasing. Monotone movement of prices is preferable in iterative auctions since it makes easier to forecast the outcome of equilibrium price computation. For the *unit-demand* auction model where each bidder desires at most one item, an ascending auction and a descending auction are proposed by Demange–Gale–Sotomayor [5] and by Mishra–Parkes [9], respectively. These iterative auctions are generalized to the multi-demand auction model by Kelso–Crawford [7], Gul–Stacchetti [6], and Ausubel [2].

While ascending and descending auctions have a merit that the price movement is monotone, these iterative auctions have a drawback that the number of iterations is large. In an ascending auction, we cannot decrease prices during the computation, and therefore the initial prices should be lower bounds of unknown equilibrium prices. Therefore, even if the auctioneer knows the expectation of equilibrium prices, it is difficult to reduce the number of iterations by using the knowledge. It is customary in ascending auctions to set the initial prices to the lowest possible prices, but this makes the number of iterations large. This is also the case with descending auctions.

To make it possible to start from arbitrarily chosen prices, Andersson–Erlanson [1] proposed an iterative auction, for the unit-demand model, by combining the ascending auction by [5] and the descending auction by [9]. The algorithm admits the use of arbitrarily chosen initial prices, and consists of two phases, the price ascending phase and descending phase. Hence, the movement of prices is first monotone increasing and then monotone decreasing. Moreover, the flexibility in the choice of initial prices is useful in reducing the number of iterations, especially when the auctioneer has information about the expected equilibrium prices. Andersson–Erlanson [1] also theoretically analyzed the number of iterations in the two-phase auction algorithm.

The connection between equilibrium price computation and optimization is made clear in Ausubel [2], which shows that a price vector $p \in \mathbb{Z}^n_+$ is an equilibrium price vector if and only if p is a minimizer of the *Lyapunov function* $L : \mathbb{Z}^n_+ \to \mathbb{Z}$ given by

$$L(p) = \sum_{i=1}^{m} \max\{f_i(X) - p(X) \mid X \subseteq N\} + p(N) \quad (p \in \mathbb{Z}^n_+), \tag{3}$$

under the assumption that each f_i is an integer valued function satisfying the gross-substitutes condition. The connection to discrete convex analysis is pointed out in Murota–Shioura–Yang [13, 14], which show that the Lyapunov function is an $\mathrm{L}^\natural$-convex function, and hence iterative auctions can be seen as minimization algorithms for a specific $\mathrm{L}^\natural$-convex function. Indeed, it is shown [14] that some of the existing iterative auctions coincide with $\mathrm{L}^\natural$-convex function minimization algorithms applied to the Lyapunov function. In particular, the two-phase auction algorithm for the unit-demand model by Andersson–Erlanson [1] can be recognized as the algorithm TwoPhaseMinMin applied to the Lyapunov function. Moreover, for the multi-demand model, a two-phase auction algorithm is proposed in [13, 14] (see also Ausubel [2]) by applying TwoPhaseMinMin to the Lyapunov function. Hence, our result (Theorem 5) provides tight bounds for the numbers of iterations required by the two-phase auction algorithms. We here rephrase Theorem 5 for the two-phase auction in [14].

Corollary 6. *For an arbitrarily chosen initial price vector $p^\circ \in \mathbb{Z}^n_+$, the two-phase auction of Murota–Shioura–Yang* [14] *terminates by outputting the unique minimal equilibrium price vector $p^*_{\min}$. The number*

of updates of vector p is bounded by $\mu(p^\circ)$ in the ascending phase and by $\|p^\circ - p^*_{\min}\|^+_\infty$ in the descending phase; in total, bounded by $\mu(p^\circ) + \|p^\circ - p^*_{\min}\|^+_\infty$.

3 Proofs

In this section we give proofs of Theorems 1 and 5.

3.1 Proof of Theorem 1

We prove Theorem 1 by improving the analysis of a more general minimization algorithm by Kolmogorov–Shioura [8] (named "primal algorithm" in [8]) to Theorem 7 below. In this algorithm, vector p can move upwards or downwards arbitrarily in each iteration, as far as the function value $g(p)$ decreases.

Algorithm GreedyUpDown
Step 0: Find a vector $p^\circ \in \operatorname{dom} g$ and set $p := p^\circ$.
Set SuccessUp := false, SuccessDown := false.
Step 1: Do UP or DOWN in any order until SuccessUp = SuccessDown = true:
UP (do only if SuccessUp is false):
Find a minimizer $X \subseteq N$ of $g(p + \chi_X)$.
If $g(p + \chi_X) = g(p)$, then set SuccessUp := true; otherwise set $p := p + \chi_X$.
DOWN (do only if SuccessDown is false):
Find a minimizer $X \subseteq N$ of $g(p - \chi_X)$.
If $g(p - \chi_X) = g(p)$, then set SuccessDown := true; otherwise set $p := p - \chi_X$.
Step 2: Output p and stop.

The algorithm TwoPhase is a special implementation of the algorithm GreedyUpDown, where UP is performed repeatedly until SuccessUp is true, and then DOWN is performed repeatedly. We will show the following bounds for GreedyUpDown, where we say that an update of p in GreedyUpDown is an *up-update* if p is updated to $p + \chi_X$ with some nonempty X, and a *down-update* if p is updated to $p - \chi_X$ with some nonempty X.

Theorem 7. *Suppose that the algorithm* GreedyUpDown *is applied to an $L^\natural$-convex function g with an initial vector $p^\circ \in \operatorname{dom} g$. Then, the numbers of up-updates and down-updates of vector p are each bounded by $\mu(p^\circ)$; in total, the number of updates of p is bounded by $2\mu(p^\circ)$.*

Theorem 1 is an immediate corollary of this theorem.

In the following, we prove Theorem 7 by using the following property of $L^\natural$-convex functions. For a vector $q \in \mathbb{Z}^n$, we denote $\operatorname{supp}^+(q) = \{j \in N \mid q(j) > 0\}$.

Lemma 8 ([10, Theorem 7.7])**.** *Let $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ be an $L^\natural$-convex function. For every $p, q \in \operatorname{dom} g$ with $\operatorname{supp}^+(p - q) \neq \emptyset$, it holds that*

$$g(p) + g(q) \geq g(p - \chi_Y) + g(q + \chi_Y)$$

with $Y = \arg\max_{j \in N}\{p(j) - q(j)\}$.

We denote by $p^\oplus$ the unique minimal vector in $\arg\min\{g(q) \mid q \in \mathbb{Z}^n,\ q \geq p^\circ\}$, and by $p^\ominus$ the unique maximal vector in $\arg\min\{g(q) \mid q \in \mathbb{Z}^n,\ q \leq p^\circ\}$. The number of updates of vector p in the algorithm GreedyUpDown is analyzed in [8].

Proposition 9 ([8, Section 2.2])**.** *The number of up-updates (resp., down-updates) of vector p in the algorithm* GreedyUpDown *is bounded by $\|p^\oplus - p^\circ\|_\infty$ (resp., $\|p^\ominus - p^\circ\|_\infty$).*

The following lemma is a key observation to recast this result to Theorem 7.

Lemma 10. *It holds that* $\|p^{\oplus} - p^{\circ}\|_{\infty} \leq \mu(p^{\circ})$ *and* $\|p^{\ominus} - p^{\circ}\|_{\infty} \leq \mu(p^{\circ})$.

Proof. We prove the former inequality $\|p^{\oplus} - p^{\circ}\|_{\infty} \leq \mu(p^{\circ})$ only since the latter inequality $\|p^{\ominus} - p^{\circ}\|_{\infty} \leq \mu(p^{\circ})$ can be proven in the same manner.

To prove the inequality $\|p^{\oplus} - p^{\circ}\|_{\infty} \leq \mu(p^{\circ})$, we show that $\|p^{\oplus} - p^{\circ}\|_{\infty} \leq \eta(p^*, p^{\circ})$ holds for every minimizer p^* of g; recall the definition of $\eta(p^*, p^{\circ})$ in (2).

We first consider the case with $p^* \geq p^{\circ}$. Since $p^{\oplus}$ is a minimizer of g in the set $\{q \in \mathbb{Z}^n \mid q \geq p^{\circ}\}$, we have $p^{\oplus} \in \arg\min g$. Hence, it holds that

$$\|p^{\oplus} - p^{\circ}\|_{\infty} \leq \|p^* - p^{\circ}\|_{\infty} = \eta(p^*, p^{\circ}).$$

We then assume that $\mathrm{supp}^+(p^{\circ} - p^*) \neq \emptyset$. This implies $\mathrm{supp}^+(p^{\oplus} - p^*) \neq \emptyset$. Let $Y = \arg\max_{j \in N}\{p^{\oplus}(j) - p^*(j)\}$.

Claim: There exists some $t \in Y$ such that $p^{\oplus}(t) = p^{\circ}(t)$.
[Proof of Claim] By Lemma 8, it holds that

$$g(p^{\oplus}) + g(p^*) \geq g(p^{\oplus} - \chi_Y) + g(p^* + \chi_Y). \tag{4}$$

Since p^* is a minimizer of g, we have $g(p^* + \chi_Y) \geq g(p^*)$, which, combined with (4), implies $g(p^{\oplus} - \chi_Y) \leq g(p^{\oplus})$. From this inequality follows that $p^{\oplus} - \chi_Y \not\geq p^{\circ}$ since $p^{\oplus}$ is the minimal vector in the set $\arg\min\{g(q) \mid q \in \mathbb{Z}^n,\ q \geq p^{\circ}\}$. Hence, there exists some $t \in Y$ with $p^{\oplus}(t) = p^{\circ}(t)$.
[End of Proof of Claim]

It holds that

$$\begin{aligned}
\|p^* - p^{\circ}\|_{\infty}^{-} &= \max_{j \in N}\{p^{\circ}(j) - p^*(j)\} && (\text{by } \mathrm{supp}^+(p^{\circ} - p^*) \neq \emptyset) \\
&\geq p^{\circ}(t) - p^*(t) \\
&= p^{\oplus}(t) - p^*(t) && (\text{by the claim above}) \\
&= \max_{j \in N}\{p^{\oplus}(j) - p^*(j)\} && (\text{by } t \in Y \text{ and the definition of } Y).
\end{aligned}$$

It follows that for every $k \in N$, we have

$$\begin{aligned}
p^{\oplus}(k) - p^{\circ}(k) &= [p^*(k) - p^{\circ}(k)] + [p^{\oplus}(k) - p^*(k)] \\
&\leq \|p^* - p^{\circ}\|_{\infty}^{+} + \max_{j \in N}\{p^{\oplus}(j) - p^*(j)\} \\
&= \|p^* - p^{\circ}\|_{\infty}^{+} + \|p^* - p^{\circ}\|_{\infty}^{-} = \eta(p^*, p^{\circ}).
\end{aligned}$$

Hence, $\|p^{\oplus} - p^{\circ}\|_{\infty} \leq \eta(p^*, p^{\circ})$ holds. □

Theorem 7 follows from Proposition 9 and Lemma 10.

3.2 Proof of Theorem 5

The bound $\mu(p^{\circ})$ for the up phase of TwoPhaseMinMin follows from Corollary 2 since the behavior of the up phase in TwoPhaseMinMin is exactly the same as that in TwoPhaseMinMax. Hence, it suffices to prove the bound $\|p^{\circ} - p^*_{\min}\|_{\infty}^{+}$ in the down phase of TwoPhaseMinMin.

The following facts are known for TwoPhaseMinMin.

Proposition 11 ([14, Lemma 4.18])**.** *It holds that* $p^{\oplus} \geq p^*_{\min}$. *In particular, if the initial vector* p° *satisfies* $p^{\circ} \leq p^*_{\min}$, *then* $p^{\oplus} = p^*_{\min}$ *holds*

Proposition 12 ([14, Lemma 4.18])**.** *At the end of the up phase,* $p = p^{\oplus}$ *holds. In particular, if the initial vector* p° *satisfies* $p^{\circ} \leq p^*_{\min}$, *then* $p = p^*_{\min}$ *holds.*

Proposition 13 (cf. [14, Theorem 4.11]). *The number of updates of p in the down phase of* TwoPhaseMinMin *is exactly equal to $\|p^{\oplus} - p^*_{\min}\|^+_\infty$. In particular, if the initial vector p° satisfies $p^\circ \leq p^*_{\min}$, then the down phase terminates immediately without updating p.*

We show in the following lemma that the value $\|p - p^*_{\min}\|^+_\infty$ remains the same in each iteration of the up phase. Hence, we have

$$\|p^{\oplus} - p^*_{\min}\|^+_\infty = \|p^\circ - p^*_{\min}\|^+_\infty,$$

implying that the number of updates of p in the down phase is equal to $\|p^\circ - p^*_{\min}\|^+_\infty$.

By Proposition 13, we may assume that the initial vector p° satisfies $\operatorname{supp}^+(p^\circ - p^*_{\min}) \neq \emptyset$. This assumption implies that $\operatorname{supp}^+(p - p^*_{\min}) \neq \emptyset$ in each iteration of the up phase.

Lemma 14. *Let $p \in \operatorname{dom} g$, and suppose that $\operatorname{supp}^+(p - p^*_{\min}) \neq \emptyset$. Also, let $X \subseteq N$ be the minimal minimizer of $g(p + \chi_X)$. Then,*

$$\|(p + \chi_X) - p^*_{\min}\|^+_\infty = \|p - p^*_{\min}\|^+_\infty.$$

This lemma follows immediately from the following (slightly) stronger result.

Lemma 15. *Let $p \in \operatorname{dom} g$ and $p^* \in \arg\min g$, and suppose that $\operatorname{supp}^+(p - p^*) \neq \emptyset$. Also, let $X \subseteq N$ be the minimal minimizer of $g(p + \chi_X)$. Then,*

$$\|(p + \chi_X) - p^*\|^+_\infty = \|p - p^*\|^+_\infty.$$

Proof. Let $Z = \arg\max\{p(j) - p^*(j) \mid j \in N\}$. If $Z \cap X = \emptyset$, then we have the desired equation $\|(p + \chi_X) - p^*\|^+_\infty = \|p - p^*\|^+_\infty$. Assume, to the contrary, that $Z \cap X \neq \emptyset$. Then, we have

$$\arg\max\{(p + \chi_X)(i) - p^*(i) \mid i \in N\} = Z \cap X.$$

Hence, Lemma 8 implies that

$$\begin{aligned} g(p + \chi_X) + g(p^*) &\geq g(p + \chi_X - \chi_{Z\cap X}) + g(p^* + \chi_{Z\cap X}) \\ &= g(p + \chi_{X\setminus Z}) + g(p^* + \chi_{Z\cap X}). \end{aligned}$$

We have $g(p^* + \chi_{Z\cap X}) \geq g(p^*)$ since p^* is a minimizer of g. Hence, it holds that $g(p + \chi_{X\setminus Z}) \leq g(p + \chi_X)$, a contradiction to the minimality of X. □

This concludes the proof of Theorem 5.

References

[1] Andersson, T., Erlanson, A.: Multi-item Vickrey–English–Dutch auctions. Games Econ. Behav. **81**, 116–129 (2013)

[2] Ausubel, L.M.: An efficient dynamic auction for heterogeneous commodities. Amer. Econ. Rev. **96**, 602–629 (2006)

[3] Blumrosen, L., Nisan, N.: Combinatorial auction. In: Nisan, N., Roughgarden, T., Tardos, É., Vazirani, V.V. (eds.) Algorithmic Game Theory, pp. 267–299. Cambridge Univ. Press, Cambridge (2007)

[4] Cramton, P., Shoham, Y., Steinberg, R.: Combinatorial Auctions. MIT Press, Cambridge, MA (2006)

[5] Demange, G., Gale, D., Sotomayor, M.: Multi-item auctions. J. Polit. Econ. **94**, 863–872 (1986)

[6] Gul, F., Stacchetti, E.: The English auction with differentiated commodities. J. Econom. Theory **92**, 66–95 (2000)

[7] Kelso Jr., A.S., Crawford, V.P.: Job matching, coalition formation, and gross substitutes. Econometrica **50**, 1483–1504 (1982)

[8] Kolmogorov, V., Shioura, A.: New algorithms for convex cost tension problem with application to computer vision. Discrete Optim. **6**, 378–393 (2009)

[9] Mishra, D., Parkes, D.C.: Multi-item Vickrey–Dutch auctions. Games Econ. Behav. **66**, 326–347 (2009)

[10] Murota, K.: Discrete Convex Analysis. SIAM, Philadelphia (2003)

[11] Murota, K.: Discrete convex analysis: A tool for economics and game theory. J. Mech. Inst. Design **1**, 151–273 (2016)

[12] Murota, K., Shioura, A.: Exact bounds for steepest descent algorithms of L-convex function minimization. Oper. Res. Lett. **42**, 361–366 (2014)

[13] Murota, K., Shioura, A., Yang, Z.: Computing a Walrasian equilibrium in iterative auctions with multiple differentiated items. Extended abstract version in: Cai, L., Cheng, S.-W., Lam, T.W. (eds.) Proceedings of the 24th International Symposium on Algorithms and Computation (ISAAC 2013), LNCS **8283**, pp. 468–478. Springer, Berlin (2013); Full paper version in: Technical Report METR 2013-10, University of Tokyo (2013).

[14] Murota, K., Shioura, A., Yang, Z.: Time bounds for iterative auctions: a unified approach by discrete convex analysis. Discrete Optim. **19**, 36–62 (2016)

[15] Shioura, A.: Algorithms for L-convex function minimization: connection between discrete convex analysis and other research fields. J. Oper. Res. Soc. Japan **60**, (2017), to appear.

Derandomization for monotone k-submodular maximization

Hiroki Oshima

Department of Mathematical Informatics,
Graduate School of Information Science and
Technology,
University of Tokyo,
Tokyo, 113-8656, Japan
hiroki_oshima@mist.i.u-tokyo.ac.jp

Abstract: Submodularity is one of the most important property of combinatorial optimization, and k-submodularity is a generalization of submodularity. Maximization of a k-submodular function is NP-hard, and approximation algorithm has been studied. For monotone k-submodular functions, [Iwata, Tanigawa, and Yoshida 2016] gave $k/(2k-1)$-approximation algorithm. In this paper, we give a deterministic algorithm by derandomizing that algorithm. Our algorithm is $k/(2k-1)$-approximation and runs in polynomial time.

Keywords: **k-submodular function, approximation algorithm, derandomization**

1 Introduction

A set function $f : 2^V \to \mathbb{R}$ is submodular if, for any $A, B \subseteq V$, $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$. Submodularity is one of the most important properties of combinatorial optimization. The rank functions of matroids and cut capacity functions of networks are submodular. Submodular functions can be seen as discrete version of convex functions.

For submodular function minimization, Grötschel et al. [4] showed the first polynomial-time algorithm. Combinatorial strongly polynomial algorithms were shown by Iwata et al. [6] and Schrijver [9]. On the other hand, submodular function maximization is NP-hard and we can only use approximation algorithms. Let an input function for maximization be f, a maximizer of f be S^*, and an output of an algorithm be S. The approximation ratio of the algorithm is defined as $f(S)/f(S^*)$ for deterministic algorithms and $\mathbb{E}[f(S)]/f(S^*)$ for randomized algorithms. A randomized version of the Double Greedy algorithm [2] achieves 1/2-approximation. Feige et al. [3] showed that $(1/2 + \epsilon)$-approximation requires exponential number of value oracle queries. This implies that, the randomized Double Greedy algorithm is one of the best algorithms in terms of the approximation ratio. Buchbinder and Feldman [1] showed a derandomized version of the randomized Double Greedy algorithm, and their algorithm achieves 1/2-approximation.

k-submodularity is an extension of submodularity. It was first introduced by Huber and Kolmogolov [5]. k-submodular function is defined as below.

Definition 1 ([5]) *Let* $(k+1)^V := \{(X_1, ..., X_k) \mid X_i \subseteq V \ (i = 1, ..., k), X_i \cap X_j = \emptyset \ (i \neq j)\}$. *A function* $f : (k+1)^V \to \mathbb{R}$ *is called k-submodular if we have*

$$f(\boldsymbol{x}) + f(\boldsymbol{y}) \geq f(\boldsymbol{x} \sqcap \boldsymbol{y}) + f(\boldsymbol{x} \sqcup \boldsymbol{y})$$

for any $\boldsymbol{x} = (X_1, ..., X_k), \ \boldsymbol{y} = (Y_1, ..., Y_k) \in (k+1)^V$. *Note that*

$$\begin{aligned} \boldsymbol{x} \sqcap \boldsymbol{y} &= (X_1 \cap Y_1, ..., X_k \cap Y_k) \ \ \textit{and} \\ \boldsymbol{x} \sqcup \boldsymbol{y} &= (X_1 \cup Y_1 \setminus \bigcup_{i \neq 1} (X_i \cup Y_i), ..., X_k \cup Y_k \setminus \bigcup_{i \neq k} (X_i \cup Y_i)). \end{aligned}$$

It is a submodular function if $k = 1$. It is called a bisubmodular function if $k = 2$.

Maximization for k-submodular functions is also NP-hard and approximation algorithms have been studied. An input of the problem is a nonnegative k-submodular function. Note that, for any k-submodular function f and any $c \in \mathbb{R}$, a function $f'(\boldsymbol{x}) := f(\boldsymbol{x}) + c$ is k-submodular. The input function is accessed via value oracle queries. An output of the problem is $\boldsymbol{x} = (X_1, ..., X_k) \in (k+1)^V$. Let an input k-submodular function be f, a maximizer of f be $\boldsymbol{o}$, and an output of an algorithm be $\boldsymbol{s}$. Then we define the approximation ratio of the algorithm as $f(\boldsymbol{s})/f(\boldsymbol{o})$ for deterministic algorithms, and $\mathbb{E}[f(\boldsymbol{s})]/f(\boldsymbol{o})$ for randomized algorithms. For bisubmodular functions, Iwata et al. [7] and Ward and Živný [10] showed that the algorithm for submodular functions [2] can be extended. Ward and Živný [10] analyzed an extension for k-submodular functions. They showed a randomized $1/(1+a)$-approximation algorithm with $a = max\{1, \sqrt{(k-1)/4}\}$ and a deterministic 1/3-approximation algorithm. Now we have a 1/2-approximation algorithm shown by Iwata et al. [8]. In particular, for monotone k-submodular functions, they gave a $\frac{k}{2k-1}$-approximation algorithm. They also showed any $(\frac{k+1}{2k} + \epsilon)$-approximation algorithm requires exponential number of value oracle queries.

In this paper, we give a deterministic approximation algorithm for monotone k-submodular maximization. It satisfies $\frac{k}{2k-1}$-approximation and runs in polynomial-time. Our algorithm is a derandomized version of the algorithm for monotone functions [8]. We also note our derandomization is an extension of the scheme used for the Double Greedy algorithm [1].

2 Preliminary

Define a partial order $\preceq$ on $(k+1)^V$ for $\boldsymbol{x} = (X_1, ..., X_k)$ and $\boldsymbol{y} = (Y_1, ..., Y_k)$ as follows:

$$\boldsymbol{x} \preceq \boldsymbol{y} \overset{\text{def}}{\iff} X_i \subseteq Y_i (i = 1, ..., k).$$

Also, for $\boldsymbol{x} = (X_1, ..., X_k) \in (k+1)^V$, $e \notin \bigcup_{l=1}^k X_l$, and $i \in \{1, ..., k\}$, define

$$\Delta_{e,i} f(\boldsymbol{x}) = f(X_1, ..., X_{i-1}, X_i \cup \{e\}, X_{i+1}, ..., X_k) - f(X_1, ..., X_k).$$

A monotone k-submodular function is k-submodular and satisfies $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$ for any $\boldsymbol{x} = (X_1, ..., X_k)$ and $\boldsymbol{y} = (Y_1, ..., Y_k)$ in $(k+1)^V$ with $\boldsymbol{x} \preceq \boldsymbol{y}$.

The property of k-submodularity can be written in another form.

Theorem 2 ([10] THEOREM 7) *A function $f : (k+1)^V \to \mathbb{R}$ is k-submodular if and only if f is orthant submodular and pairwise monotone.*

Note that orthant submodularity is to satisfy

$$\Delta_{e,i} f(\boldsymbol{x}) \geq \Delta_{e,i} f(\boldsymbol{y}) \ \ (\boldsymbol{x}, \boldsymbol{y} \in (k+1)^V, \ \boldsymbol{x} \preceq \boldsymbol{y}, \ e \notin \bigcup_{l=1}^{k} Y_l, \ i \in \{1, ..., k\}),$$

and pairwise monotonicity is to satisfy

$$\Delta_{e,i} f(\boldsymbol{x}) + \Delta_{e,j} f(\boldsymbol{x}) \geq 0 \ (\boldsymbol{x} \in (k+1)^V, \ e \notin \bigcup_{l=1}^{k} X_l, \ i, j \in \{1, ..., k\} \ (i \neq j)).$$

To analyze k-submodular functions, it is often convenient to identify $(k+1)^V$ as $\{0, 1, ..., k\}^V$. A $|V|$-dimensional vector $\boldsymbol{x} \in \{0, 1, ..., k\}^V$ is associated with $(X_1, ..., X_n) \in (k+1)^V$ by $X_i = \{e \in V \mid \boldsymbol{x}(e) = i\}$.

3 Existing randomized algorithms

3.1 Algorithm framework

In this section, we see the framework to maximize k-submodular functions (Algorithm 1 [8]). Iwata et al. [7] and Ward and Živný [10] used it with specific distributions.

Algorithm 1 ([8] Algorithm 1)

Input: A nonnegative k-submodular function $f : \{0, 1, ..., k\}^V \to \mathbb{R}_+$.
Output: A vector $\boldsymbol{s} \in \{0, 1, ..., k\}^V$.
$\boldsymbol{s} \leftarrow \boldsymbol{0}$.
Denote the elements of V by $e^{(1)}, ..., e^{(n)}$ ($|V| = n$).
for $j = 1, ..., n$ **do**
  Set a probability distribution $p^{(j)}$ over $\{1, ..., k\}$.
  Let $\boldsymbol{s}(e^{(j)}) \in \{1, ..., k\}$ be chosen randomly with $\Pr[\boldsymbol{s}(e^{(j)}) = i] = p_i^{(j)}$.
end for
return $\boldsymbol{s}$

Algorithm 1 is not only used for monotone functions. However, in this paper, we only use it for monotone functions.

Now we define some variables to analyze Algorithm 1. Let $\boldsymbol{o}$ be an optimal solution, and we write $\boldsymbol{s}^{(j)}$ as $\boldsymbol{s}$ at the j-th iteration. Let other variables be as follows:

$$\boldsymbol{o}^{(j)} = (\boldsymbol{o} \sqcup \boldsymbol{s}^{(j)}) \sqcup \boldsymbol{s}^{(j)} \quad , \quad \boldsymbol{t}^{(j-1)}(e) = \begin{cases} \boldsymbol{o}^{(j)}(e) & (e \neq e^{(j)}) \\ 0 & (e = e^{(j)}) \end{cases}$$
$$y_i^{(j)} = \Delta_{e^{(j)},i} f(\boldsymbol{s}^{(j-1)}) \quad , \quad a_i^{(j)} = \Delta_{e^{(j)},i} f(\boldsymbol{t}^{(j-1)})$$

Algorithm 1 satisfies following lemma.

Lemma 3 ([8] LEMMA 2.1.)
Let $c \in \mathbb{R}_+$. Conditioning on $\boldsymbol{s}^{(j-1)}$, suppose that

$$\sum_{i=1}^{k} (a_{i^*}^{(j)} - a_i^{(j)}) p_i^{(j)} \leq c \sum_{i=1}^{k} (y_i^{(j)} p_i^{(j)})$$

holds for each j with $1 \leq j \leq n$, where $i^* = \boldsymbol{o}(e^{(j)})$. Then $\mathbb{E}[f(\boldsymbol{s})] \geq \frac{1}{1+c} f(\boldsymbol{o})$.

3.2 A randomized algorithm for monotone functions

In this section, we see the randomized $\frac{k}{2k-1}$-approximation algorithm for monotone functions [8]. We show their algorithm as Algorithm 2.

Algorithm 2 runs in polynomial time. The approximation ratio of Algorithm 2 satisfies the theorem below.

Theorem 4 ([8] THEOREM 2.2.) *Let $\boldsymbol{o}$ be a maximizer of a monotone k-submodular function f and let $\boldsymbol{s}$ be the output of Algorithm 2. Then $\mathbb{E}[f(\boldsymbol{s})] \geq \frac{k}{2k-1} f(\boldsymbol{o})$.*

In the proof of this theorem (see [8]), the inequality of Lemma 3 is proved with $c = 1 - \frac{1}{k}$. We get $a_i \geq 0$ ($\forall i \in \{1, ..., k\}$) from monotonicity, and $a_i \leq y_i$ ($\forall i \in \{1, ..., k\}$) from orthant submodularity. Hence, the inequality

$$\sum_{i \neq i^*} (y_{i^*}^{(j)} p_i^{(j)}) \leq \left(1 - \frac{1}{k}\right) \sum_{i=1}^{k} (y_i^{(j)} p_i^{(j)}) \tag{1}$$

is used. The inequality of Lemma 3 is satisfied when the inequality (1) is valid.

Algorithm 2 ([8] Algorithm 3)

Input: A monotone k-submodular function $f : \{0, 1, ..., k\}^V \to \mathbb{R}_+$.
Output: A vector $\boldsymbol{s} \in \{0, 1, ..., k\}^V$.
$\boldsymbol{s} \leftarrow \mathbf{0}, t \leftarrow k - 1$.
Denote the elements of V by $e^{(1)}, ..., e^{(n)}$ $(|V| = n)$.
for $j = 1, ..., n$ **do**
  $y_i^{(j)} \leftarrow \Delta_{e^{(j)},i} f(\boldsymbol{s})$ $(1 \le i \le k)$.
  $\beta \leftarrow \sum_{i=1}^{k} (y_i^{(j)})^t$.
  if $\beta \neq 0$ **then** $p_i^{(j)} \leftarrow (y_i^{(j)})^t / \beta$ $(1 \le i \le k)$.
  else
    $p_1^{(j)} = 1, p_i^{(j)} = 0$ $(i = 2, ..., k)$.
  end if
  Let$\boldsymbol{s}(e^{(j)}) \in \{1, ..., k\}$ be chosen randomly with $\Pr[\boldsymbol{s}(e^{(j)}) = i] = p_i^{(j)}$.
end for
return $\boldsymbol{s}$

4 Deterministic algorithm

In this section, we give a polynomial-time deterministic algorithm for maximizing monotone k-submodular functions. Our algorithm is Algorithm 3. Algorithm 3 is a derandomized version of Algorithm 2. We note the derandomization of this algorithm is an extension of the scheme used for submodular maximization [1].

In the algorithm, we construct a distribution $\mathcal{D}$ which satisfies $\mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}}[f(\boldsymbol{s})] \ge \frac{k}{2k-1} f(\boldsymbol{o})$. Then the algorithm outputs the best solution in $\mathrm{supp}(\mathcal{D}) := \{\boldsymbol{s} \mid (p, \boldsymbol{s}) \in \mathcal{D}\}$. We can see the right hand side of (2) in Algorithm 3 is the expected value of the left hand side of (1) for $\boldsymbol{s} \sim \mathcal{D}_{j-1}$ with $i^* = l$. This is because $\sum_{i \neq l} p_{i,\boldsymbol{s}} y_l(\boldsymbol{s}) = (1 - p_{l,\boldsymbol{s}}) y_l(\boldsymbol{s})$. Also the left hand side of (2) is the expected value of the right hand side of (1) with $c = 1 - 1/k$. From (3) and (4), $\mathcal{D}_j$ in (5) is constructed as a distribution.

Algorithm 3 achieves the same approximation ratio as Algorithm 2.

Theorem 5 *Let $\boldsymbol{o}$ be a maximizer of a monotone nonnegative k-submodular function f and let $\boldsymbol{z}$ be the output of Algorithm 3. Then $f(\boldsymbol{z}) \ge \frac{k}{2k-1} f(\boldsymbol{o})$.*

Proof: We consider the j-th iteration. From (5), we get

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[\sum_{i=1}^{k} p_{i,\boldsymbol{s}} y_i(\boldsymbol{s})\right] &= \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[\sum_{i=1}^{k} p_{i,\boldsymbol{s}} (f(\boldsymbol{s}_{e^{(j)},i}) - f(\boldsymbol{s}))\right] \\
&= \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[\sum_{i=1}^{k} p_{i,\boldsymbol{s}} f(\boldsymbol{s}_{e^{(j)},i}) - f(\boldsymbol{s})\right] \\
&= \mathbb{E}_{\boldsymbol{s}'\sim\mathcal{D}_j}\left[f(\boldsymbol{s}')\right] - \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[f(\boldsymbol{s})\right]. \qquad (6)
\end{aligned}$$

Now, we consider $\boldsymbol{o}[\boldsymbol{s}] := (\boldsymbol{o} \sqcup \boldsymbol{s}) \sqcup \boldsymbol{s}$. Define the variables as follows:

$$\begin{aligned}
\boldsymbol{r}(e) &= \begin{cases} \boldsymbol{o}[\boldsymbol{s}](e) & (e \neq e^{(j)}) \\ 0 & (e = e^{(j)}) \end{cases} \\
a_i(\boldsymbol{s}) &= \Delta_{e^{(j)},i} f(\boldsymbol{r})
\end{aligned}$$

Then we have

$$f(\boldsymbol{o}[\boldsymbol{s}]) - f(\boldsymbol{o}[\boldsymbol{s}_{e^{(j)},i}]) = a_{i^*}(\boldsymbol{s}) - a_i(\boldsymbol{s}) \quad (i^* = \boldsymbol{o}(e^{(j)})) \qquad (7)$$

Algorithm 3 A deterministic algorithm

Input: A monotone k-submodular function $f : \{0, 1, ..., k\}^V \to \mathbb{R}_+$.
Output: A vector $\boldsymbol{s} \in \{0, 1, ..., k\}^V$.
$\mathcal{D}_0 \leftarrow (1,\ \mathbf{0}),\ (\mathcal{D} = \{(p, \boldsymbol{s}) \mid \boldsymbol{s} \in (k+1)^V,\ 0 \leq p \leq 1\}\ (\sum_{\boldsymbol{s}\in\mathcal{D}} p = 1))$.
Denote the elements of V by $e^{(1)}, ..., e^{(n)}$ $(|V| = n)$.
for $j = 1, ..., n$ **do**
$y_i(\boldsymbol{s}) \leftarrow \Delta_{e^{(j)},i} f(\boldsymbol{s})$ $(\forall \boldsymbol{s} \in \mathrm{supp}(\mathcal{D}_{j-1}),\ i \in \{1, ..., k\})$.
Find an extreme point solution $(p_{i,\boldsymbol{s}})_{i=1,...,k,\ \boldsymbol{s}\in\mathrm{supp}(\mathcal{D}_{j-1})}$of the following linear formulation:

$$\left(1 - \frac{1}{k}\right) \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}} \left[\sum_{i=1}^{k} p_{i,\boldsymbol{s}} y_i(\boldsymbol{s})\right] \geq \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}} [(1 - p_{l,\boldsymbol{s}}) y_l(\boldsymbol{s})] \quad (2)$$

$$(l \in \{1, ..., k\})$$

$$\sum_{i=1}^{k} p_{i,\boldsymbol{s}} = 1 \quad (\forall \boldsymbol{s} \in \mathrm{supp}(\mathcal{D}_{j-1})) \quad (3)$$

$$p_{i,\boldsymbol{s}} \geq 0 \quad (\forall \boldsymbol{s} \in \mathrm{supp}(\mathcal{D}_{j-1}),\ i \in \{1, ..., k\}). \quad (4)$$

Construct a new distribution $\mathcal{D}_j$:

$$\mathcal{D}_j \leftarrow \bigcup_{i=1}^{k} \{(p_{i,\boldsymbol{s}} \cdot \Pr_{\mathcal{D}_{j-1}}[\boldsymbol{s}],\ \boldsymbol{s}_{e^{(j)},i}) \mid \boldsymbol{s} \in \mathrm{supp}(\mathcal{D}_{j-1}),\ p_{i,\boldsymbol{s}} > 0\} \quad (5)$$

$$\left(\boldsymbol{s}_{e^{(j)},i}(e) = \begin{cases} \boldsymbol{s}(e) & (e \neq e^{(j)}) \\ i & (e = e^{(j)}) \end{cases}\right).$$

end for
return $\arg\max_{\boldsymbol{s}\in\mathrm{supp}(\mathcal{D}_n)} \{f(\boldsymbol{s})\}$

From monotonicity and orthant submodularity of f, we have

$$a_{i^*}(\boldsymbol{s}) - a_i(\boldsymbol{s}) \leq y_{i^*}(\boldsymbol{s}). \tag{8}$$

From (7) and (8), we get

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[f(\boldsymbol{o}[\boldsymbol{s}])\right] - \mathbb{E}_{\boldsymbol{s}'\sim\mathcal{D}_j}\left[f(\boldsymbol{o}[\boldsymbol{s}'])\right] &= \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[\sum_{i=1}^{k} p_{i,\boldsymbol{s}} f(\boldsymbol{o}[\boldsymbol{s}]) - f(\boldsymbol{o}[\boldsymbol{s}_{e^{(j)},i}])\right] \\
&= \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[\sum_{i=1}^{k} p_{i,\boldsymbol{s}} \left(f(\boldsymbol{o}[\boldsymbol{s}]) - f(\boldsymbol{o}[\boldsymbol{s}_{e^{(j)},i}])\right)\right] \\
&= \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[\sum_{i\neq i^*} p_{i,\boldsymbol{s}} \left(a_{i^*}(\boldsymbol{s}) - a_i(\boldsymbol{s})\right)\right] \\
&\leq \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[\sum_{i\neq i^*} p_{i,\boldsymbol{s}} \left(y_{i^*}(\boldsymbol{s})\right)\right] \\
&= \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[\left(1 - p_{i^*,\boldsymbol{s}}\right)\left(y_{i^*}(\boldsymbol{s})\right)\right]. \qquad (9)
\end{aligned}$$

$p_{i,\boldsymbol{s}}$ satisfies (2) for all $l \in \{1, 2, ..., k\}$. Hence we obtain

$$\left(1 - \frac{1}{k}\right)\left(\mathbb{E}_{\boldsymbol{s}'\sim\mathcal{D}_j}\left[f(\boldsymbol{s}')\right] - \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[f(\boldsymbol{s})\right]\right) \geq \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_{j-1}}\left[f(\boldsymbol{o}[\boldsymbol{s}])\right] - \mathbb{E}_{\boldsymbol{s}'\sim\mathcal{D}_j}\left[f(\boldsymbol{o}[\boldsymbol{s}'])\right] \tag{10}$$

from (6) and (9). By the summation of (10), we get

$$\left(1 - \frac{1}{k}\right)\left(\mathbb{E}_{\boldsymbol{s}'\sim\mathcal{D}_n}\left[f(\boldsymbol{s}')\right] - \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_0}\left[f(\boldsymbol{s})\right]\right) \geq \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}_0}\left[f(\boldsymbol{o}[\boldsymbol{s}])\right] - \mathbb{E}_{\boldsymbol{s}'\sim\mathcal{D}_n}\left[f(\boldsymbol{o}[\boldsymbol{s}'])\right]. \tag{11}$$

Note that $\boldsymbol{o}[\boldsymbol{s}'] = \boldsymbol{s}'$ for $\boldsymbol{s}' \in \mathrm{supp}(\mathcal{D}_n)$, and $\boldsymbol{o}[\boldsymbol{s}] = \boldsymbol{o}$ for $\boldsymbol{s} \in \mathrm{supp}(\mathcal{D}_0)$. Now we have

$$\begin{aligned}
f(\boldsymbol{o}) &\leq \left(2 - \frac{1}{k}\right)\mathbb{E}_{\boldsymbol{s}'\sim\mathcal{D}_n}\left[f(\boldsymbol{s}')\right] - \left(1 - \frac{1}{k}\right)f(\boldsymbol{0}) \\
&\leq \left(2 - \frac{1}{k}\right)\mathbb{E}_{\boldsymbol{s}'\sim\mathcal{D}_n}\left[f(\boldsymbol{s}')\right] \\
&\leq \left(2 - \frac{1}{k}\right)\max_{\boldsymbol{s}'\in\mathrm{supp}(\mathcal{D}_n)}\{f(\boldsymbol{s}')\}
\end{aligned}$$

□

The algorithm performs a polynomial number of value oracle queries.

Theorem 6 *Algorithm 3 returns a solution after* $\mathrm{O}(n^2k^2)$ *value oracle queries.*

Proof: Algorithm 3 uses the value oracle to caluculate $y_i(\boldsymbol{s})$. At the j-th iteration, the number of $y_i(\boldsymbol{s})$ is $k|\mathcal{D}_{j-1}|$. From (5), $|\mathcal{D}_j|$ equals the number of $p_{i,\boldsymbol{s}} \neq 0$. Then we have to consider $p_{i,\boldsymbol{s}} \neq 0$ at the j-th iteration.

By the definition, $(p_{i,\boldsymbol{s}})_{i=1,...,k,\ \boldsymbol{s}\in\mathrm{supp}(\mathcal{D}_{j-1})}$ is an extreme point solution of (2), (3), and (4). Note that, we can get a solution by setting $(p_{i,\boldsymbol{s}})$ as the distribution of Algorithm 2 for each $\boldsymbol{s} \in \mathrm{supp}(\mathcal{D}_{j-1})$. We can also see the feasible region of (2), (3), and (4) is bounded. Then some extreme point solution exists.

Let $|\mathcal{D}_{j-1}| = m$. By $(p_{i,\boldsymbol{s}})_{i=1,...,k,\ \boldsymbol{s}\in\mathrm{supp}(\mathcal{D}_{j-1})} \in \mathbb{R}^{km}$ and k equalities of (3), $km - k$ inequalities are tight at any extreme point solution. (2) have m inequalities and (4) have km inequalities. Then, at least $km - k - m$ inequalities of (4) are tight. Hence, the number of $p_{i,\boldsymbol{s}} \neq 0$ is at most $m + k$.

Now we have $|\mathcal{D}_j| \leq |\mathcal{D}_{j-1}| + k$. We can also see $|\mathcal{D}_j| \leq jk + 1$. Then the number of value oracle queries is

$$\sum_{j=1}^{n} k|\mathcal{D}_{j-1}| \leq \sum_{j=1}^{n} k(jk+1).$$

□

In our algorithm, we have to search for an extreme point solution. We can do it by solving LP for some objective function. If we use LP for our algorithm, it is polynomial-time not only for the number of queries but also for the number of operations. The simplex method is not proved to be a polynomial-time method. However, it is practical. Our algorithm needs only an extreme point solution, then if we get a basic solution, it is enough. So we can use the first phase of two-phase simplex method to find an extreme point solution.

5 Conclusion

We showed a derandomized algorithm for monotone k-submodular maximization. It is $\frac{k}{2k-1}$-approximation and polynomial-time algorithm.

One of open problems is a faster method for finding an extreme point solution of the linear formulation. For submodular functions, Buchbinder and Feldman [1] showed greedy methods are effective. It is because their formulation is the form of fractional knapsack problem. Our formulation is similar to theirs, and ours can be seen as the form of an LP relaxation of multidimensional knapsack problem. However, faster methods are not given than general LP solutions. The number of constraints in our formulation depends on k and the number of iterations. It is therefore difficult to find an extreme point faster.

Constructing a deterministic algorithm for nonmonotone functions is also an important open problem. For nonmonotone functions, we have pairwise monotonicity instead of monotonicity. In such a situation, for some i, a_i can be negative. However, if $y_j > 0$ for all j, we can't find such i. Then, if we try to use the same derandomizing method, the number of constraints in the linear formulation and the size of $\mathcal{D}$ will be exponential. So the algorithm cannot finish in polynomial-time.

References

[1] N. Buchbinder and M. Feldman, Deterministic algorithms for submodular maximization problems, *In Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2016, 392–403

[2] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz, A tight linear time (1/2)-approximation for unconstrained submodular maximization, *SIAM Journal on Computing*, **44**(5), 2015, 1384–1402

[3] U. Feige, V. S. Mirrokni, and J. Vondrák, Maximizing non-monotone submodular functions, *SIAM Journal on Computing* **40**(4), 2011, 1133–1153

[4] M. Grötschel, L. Lovász, and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* **1**(2), 1981, 169–197

[5] A. Huber and V. Kolmogorov, Towards minimizing k-submodular functions, *In Proceedings of 2nd International Symposium on Combinatorial Optimization*, 2012, 451–462

[6] S. Iwata, L. Fleischer, and S. Fujishige, A combinatorial strongly polynomial algorithm for minimizing submodular functions, *Journal of the ACM* **48**(4), 2001, 761–777

[7] S. Iwata, S. Tanigawa, and Y. Yoshida, Bisubmodular function maximization and extensions, Technical report, Technical Report METR 2013-16, The University of Tokyo, 2013

[8] S. Iwata, S. Tanigawa, and Y. Yoshida, Improved approximation algorithms for k-submodular function maximization, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms* , SIAM, 2016, 404–413

[9] A. Schrijver, A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *Journal of Combinatorial Theory , Series B*, **80**(2), 2000, 346–355

[10] J. Ward and S. Živný, Maximizing k-submodular functions and beyond, *ACM Trans. Algorithms*, **12**(4), 2016, 47:1–47:26

Progression-free sets and the polynomial method

Péter P. Pach[1]

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
1117 Budapest, Magyar tudósok körútja 2,
Hungary
ppp@cs.bme.hu

Extended abstract

We say that a subset A of an (additively written) abelian group G is *progression-free* if there do not exist pairwise distinct $a, b, c \in A$ with $a + b = 2c$, and we denote by $r_3(G)$ the largest size of a progression-free subset $A \subseteq G$. For abelian groups G of odd order, Brown and Buhler [2] and independently Frankl, Graham, and Rödl [6] proved that $r_3(G) = o(|G|)$ as $|G|$ grows. Meshulam [10], following the general lines of Roth's argument, has shown that if G is an abelian group of odd order, then $r_3(G) \leq 2|G|/\mathrm{rk}(G)$ (where we use the standard notation $\mathrm{rk}(G)$ for the rank of G); in particular, $r_3(\mathbb{Z}_m^n) \leq 2m^n/n$. Despite many efforts, no further progress was made for over 15 years, till Bateman and Katz in their ground-breaking paper [1] proved that $r_3(\mathbb{Z}_3^n) = O(3^n/n^{1+\varepsilon})$ with an absolute constant $\varepsilon > 0$.

Abelian groups of even order were first considered in [8] where, as a further elaboration on the Roth-Meshulam proof, it is shown that $r_3(G) < 2|G|/\mathrm{rk}(2G)$ for any finite abelian group G; here $2G = \{2g \colon g \in G\}$. For the homocyclic groups of exponent 4 this result was improved by Sanders [12] who proved that $r_3(\mathbb{Z}_4^n) = O(4^n/n(\log n)^{\varepsilon})$ with an absolute constant $\varepsilon > 0$. In [3] we further improved Sanders's result, as follows.

Let H denote the binary entropy function; that is,

$$H(x) = -x \log_2 x - (1-x) \log_2 (1-x), \quad x \in (0,1),$$

where $\log_2 x$ is the base-2 logarithm of x. Let

$$\gamma := \max \left\{ \frac{1}{2} \big(H(0.5 - \varepsilon) + H(2\varepsilon) \big) \colon 0 < \varepsilon < 0.25 \right\} \approx 0.926.$$

Theorem 1 *If $n \geq 1$ and $A \subseteq \mathbb{Z}_4^n$ is progression-free, then $|A| \leq 4^{\gamma n}$.*

We note that the exponential reduction in Theorem is the first of its kind for problems of this sort.

Starting from Roth, the standard way to obtain quantitative estimates for $r_3(G)$ involves a combination of the Fourier analysis and the density increment technique; the only exception is [9] where for the groups $G \cong \mathbb{Z}_q^n$ with a prime power q, the above-mentioned Meshulam's result is recovered using a completely elementary argument. In contrast, in [3] we use the polynomial method without resorting to the familiar Fourier analysis – density increment strategy. Our result is based on the following lemma:

Lemma 2 *Suppose that $n \geq 1$ and $d \geq 0$ are integers, P is a multilinear polynomial in n variables of total degree at most d over a field $\mathbb{F}$, and $A \subseteq \mathbb{F}^n$ is a set with $|A| > 2 \sum_{0 \leq i \leq d/2} \binom{n}{i}$. If $P(a-b) = 0$ for all $a, b \in A$ with $a \neq b$, then also $P(0) = 0$.*

[1]Research is supported by the National Research, Development and Innovation Office of Hungary (Grant Nr. NKFIH (OTKA) PD115978 and NKFIH (OTKA) K108947) and the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

In the past year some interesting applications of our method were obtained including the solution of the cap set problem [4], the proof of the Erdős-Szemerédi sunflower conjecture [11], tight bound for Green's arithmetic triangle removal lemma [5], growth rate of tri-colored sumfree sets [7].
I will talk about this method and some of the applications.
This is joint work with Ernie Croot and Seva Lev.

References

[1] M. Bateman and N.H. Katz, New bounds on cap sets, *J. Amer. Math. Soc.* **25** (2012), no. 2, 585–613.

[2] T.C. Brown and J.C. Buhler, A density version of a geometric Ramsey theorem, *J. Combin. Theory, Ser. A* **32** (1982), 20–34.

[3] E. Croot, V.F. Lev, P.P. Pach, Progression-free sets in $\mathbb{Z}_4^n$ are exponentially small, *Ann. of Math.* **185 (1)** (2017) 331–337.

[4] J.S. Ellenberg, D. Gijswijt, On large subsets of $\mathbb{F}_q^n$ with no three-term arithmetic progression, *Ann. of Math.* **185 (1)** (2017) 339–343.

[5] J. Fox, L.M. Lovász, A tight bound for Green's arithmetic triangle removal lemma in vector spaces, arXiv: 1606.01230

[6] P. Frankl, G Graham, and V. Rödl, On subsets of abelian groups with no 3-term arithmetic progression, *J. Combin. Theory, Ser. A* **45** (1987), 157–161.

[7] R. Kleinberg, W.F. Sawin, D.E. Speyer, The growth-rate of tri-colored sum-free sets, arXiv: 1607.00047

[8] V.F. Lev, Progression-free sets in finite abelian groups, *J. Number Theory* **104** (2004), no. 1, 162–169.

[9] V.F. Lev, Character-free approach to progression-free sets, *Finite Fields Appl.* **18** (2012), no. 2, 378–383.

[10] R. Meshulam, On subsets of finite abelian groups with no 3-term arithmetic progressions, *J. Combin. Theory Ser. A* **71** (1995), no. 1, 168–172.

[11] E. Naslund, W.F. Sawin, Upper bounds for sunflower-free sets, arXiv: 1606.09575

[12] T. Sanders, Roth's theorem in $\mathbb{Z}_4^n$, *Anal. PDE 2* (2009), no. 2, 211–234.

Weak embeddings of posets to the Boolean lattice

Dömötör Pálvölgyi*

Department of Pure Mathematics and
Mathematical Statistics
University of Cambridge
Cambridge, UK
dom@cs.elte.hu

Abstract: The goal of this paper is to prove that several variants of deciding whether a poset can be (weakly) embedded into a small Boolean lattice, or to a few consecutive levels of a Boolean lattice, are **NP**-complete, answering a question of Griggs and of Patkós. As an equivalent reformulation of one of these problems, we also derive that it is **NP**-complete to decide whether a given graph can be embedded to the two middle levels of some hypercube. This hardness result might turn out to be an important step towards the proof of the famous **P**=**NP** conjecture - only a polynomial time algorithm for the above problem is missing.

Keywords: forbidden posets, Boolean lattice, NP-completeness, graph theory

1 Introduction

A *poset* $(P, \leq)$ is a partially ordered set on $|P|$ elements. An injective map f from poset P to poset Q is called a *weak embedding* if for every $p, q \in P$ we have $f(p) \leq f(q)$ if $p \leq q$, and it is called a *(strong) embedding* if $f(p) \leq f(q)$ if and only if $p \leq q$. Similarly, an injective map f from graph G to graph H is called an *embedding* if for any edge uv of G its image $f(u)f(v)$ is an edge of H, and it is called an *induced embedding* if uv is an edge of G if and only if $f(u)f(v)$ is an edge of H. (Be careful that simply *embedding* is *strong* embedding for posets, but for graphs, it is the equivalent of *weak* poset embeddings - unfortunately, both are standard terminology.) Sometimes it will be comfortable to use graph terminology for a poset according to its graph obtained from its Hasse diagram; the vertices of this graph are the elements of the poset, with an edge between $p < q$ if there is no r for which $p < r < q$ holds. Thus, we call the elements adjacent to an element x in the Hasse diagram the *neighbors* of x, and the length of the shortest path in the Hasse diagram connecting some elements x and y their *distance*. The *Boolean lattice* of a base set S of size n has 2^n elements, all subsets of S, where the ordering is given by containment structure, i.e., $X \leq Y$ if $X \subset Y$. The k^{th} *level* of the Boolean lattice is the collection of its elements of size k. We denote the Boolean lattice on 2^n elements by B_n and for n even, we refer to its $(\frac{n}{2})^{\text{th}}$ level as its *middle level*, while for general n, we refer to its levels from $\lfloor \frac{n-e+1}{2} \rfloor^{\text{th}}$ to $\lfloor \frac{n+e-1}{2} \rfloor^{\text{th}}$ as its e *middle levels*.

In this paper we study the decision complexity of whether a poset admits a weak embedding to (some levels of) B_n (where n is arbitrary, given as part of the input). Apparently, earlier only strong embeddings to B_n have been studied, first in [18], while the **NP**-completeness of the problem was established in [17]; for more recent results related to complexity, see [10, 15]. We find it somewhat surprising that weak embeddings have not yet been studied. There are, however, some graph problems that are equivalent to weak embedding questions to two consecutive levels, e.g., the Middle Levels conjecture is that there is a Hamiltonian cycle in the union of the two middle levels of every B_{2n+1}; this has been recently solved by Mütze [13].

*Research is supported by the Marie Skłodowska-Curie action of the EU, under grant IF 660400.

We write $P \subset Q$ if P has a weak embedding to Q. This indeed defines a partial order on the posets, i.e., $P \subset Q \subset R$ implies $P \subset R$ and $P \subset Q \subset P$ implies that P and Q are isomorphic. If $P \subset Q$, we say that Q *contains* (a copy of) P, otherwise we say that Q is P*-free*. We denote by $d(P)$ the *smallest* integer such that $P \subset B_{d(P)}$. (For strong embeddings, this parameter is called the *2-dimension* of P, and embeddings to B_n are called *bit-vector encodings*.) As $P \subset C_{|P|} \subset B_{|P|-1}$, where C_n denotes the *chain* (totally ordered poset) on n elements, $d(P)$ is always some non-negative integer. Despite the huge literature of embedding trees to the hypercube [3, 11], it seems that $d(P)$ has not even been studied for trees. The problem of determining the value of $d(T_k)$, where T_k denotes the complete binary tree of depth k, can be shown to be equivalent to a search problem proposed by G.O.H. Katona,* which is also open.

We also study weak embeddings to the union of a few consecutive levels of the Boolean lattice. We denote by $e(P)$ the *largest* integer such that any $e(P)$ consecutive levels of any Boolean lattice are P-free. It follows from the definitions that $e(P) \leq d(P)$, as any $d(P)+1$ levels of any Boolean lattice contain a copy of $B_{d(P)}$ which contains a copy of P. If P has a smallest and a largest element, then $e(P) = d(P)$, while examples for small posets for which inequality holds include the so-called *Fork* poset on three elements, a, b, c, with $a < b, c$, for which $e(P_{\text{fork}}) = 1 < d(P_{\text{fork}}) = 2$, and the so-called *Butterfly* poset on four elements, w, x, y, z, with $w, x < y, z$, for which $e(P_{\text{butterfly}}) = 2 < d(P_{\text{butterfly}}) = 3$. We also note that $h(P)-1 \leq e(P) \leq d(P)$, where $h(P)$ is the *height* of the poset, i.e., the *cardinality* of its longest subchain.

The parameter $e(P)$ has been introduced in Griggs, Li and Lu [7], as it naturally came up while studying the largest possible size of a P-free subposet of B_n, denoted by $La(n, P)$. This parameter has been first studied by Katona in the 1980s for general posets; for a recent survey see Griggs and Li [6]. The general conjecture, implicitly contained in the earlier works of Katona and others, and explicitly first stated by Bukh [2], and a couple of months later, independently, by Griggs and Lu [8], is that $\pi(P) = \lim_{n\to\infty} \frac{La(n,P)}{\binom{n}{n/2}}$ always exists, and equals to $e(P)$. (Note that $e(P) \leq \pi(P)$ follows from that the union of the $e(P)$ middle levels of B_n are P-free.) This has only been proved for special posets. The most general result is due to Bukh [2], which says that if the Hasse-diagram of P is a tree, then $\pi(P) = h(P) - 1 = e(P)$.

Motivated by this, Griggs [5] and Patkós† asked independently around the same time the complexity of determining $e(P)$.‡ Answering their questions, we show the following.

Theorem 1 *To decide whether $d(P)$ or $e(P)$ is at most n is* **NP***-complete.*

Remark 2 *In fact, as we will see from the proof, it is already* **NP***-complete for posets with a smallest and a largest element (in which case $d(P) = e(P)$) to determine whether these parameters equal $h(P)-1$.*

Theorem 3 *To decide whether $e(P) \leq 1$ is* **NP***-complete.*

Remark 4 *The graph theoretic reformulation of Theorem 3 is that it is* **NP***-complete to decide whether a given graph can be embedded to two consecutive levels of some hypercube.*

Theorem 5 *To decide whether a poset can be weakly embedded to the union of the third and fourth level of some Boolean lattice is* **NP***-complete.*

Remark 6 *Both Theorems 3 and 5 also hold for strong embeddings, as the respective posets used in their proofs can only have a strong embedding to the required structures.*

Finally, using our methods we also sketch the proof of a slightly related result.

Theorem 7 *To decide whether a graph is an induced subgraph of a Johnson graph is* **NP***-complete.*

*Emléktábla Workshop, 2013. `http://www.renyi.hu/~emlektab/emlektabla5problems.pdf`.

†Personal communication, 2014.

‡Griggs has also asked for the complexity of determining the 2-dimension of P, but this has already been proved to be **NP**-complete by Stahl and Wille [17]; for a more accessible version, see Habib et al. [10].

2 Preliminaries

2.1 Connection to graph embeddings

It is well-known that directed and undirected graph embedding problems can be easily reduced to each other by simple gadgets. * The same is true for weak poset embedding problems. To reduce a weak poset embedding problem to a directed graph embedding problem, notice that P weakly embeds to Q if and only if the *transitive closure* of P embeds to the *transitive closure* of Q. To reduce a graph embedding problem to a weak poset embedding problem, let us denote by $\hat{G}$ the two-level poset obtained from a graph G as follows. The elements of $\hat{G}$ are the vertices and edges of G, and any edge is larger than its endpoints (these are the only relations). Thus, the vertices of G form an antichain in $\hat{G}$, the lower level, and the edges of G also form an antichain in $\hat{G}$, the upper level.

Proposition 8 *G is a subgraph of H if and only if $\hat{G}$ weakly embeds to $\hat{H}$.*

The interested reader can find the simple proof of Proposition 8 in [1]. As deciding whether a graph is a subgraph of another graph, known as the SUBGRAPH ISOMORPHISM problem, is **NP**-complete [4], we get that weak embedding for posets is also **NP**-complete.

Corollary 9 *Deciding whether P weakly embeds to Q or not is* **NP***-complete, already if both P and Q have only two levels.*

Remark 10 *Note that Theorem1 is* not *a strengthening of this corollary, as here Q is also given as part of the input, while in Theorem1 B_n has exponential size (but its description, the binary encoding of n, is* $\log\log$ *of the size of B_n).*

2.2 Uniqueness of embedding two consecutive levels

Let $L_2(k)$ denote the union of the two middle levels of B_k.

Observation 11 *Any weak embedding of $L_2(k)$ to $L_2(n)$ is* distance-preserving*, i.e., the distance between any two elements of $L_2(k)$ is the same as the distance between their images in $L_2(n)$.*

Proof: The proof is by induction on k. The statement is trivially true for $k = 0$. Take a weak embedding $f : L_2(k) \to L_2(n)$. Pick an arbitrary element x of $L_2(k)$ and denote the (unique) element at distance k from it by $\bar{x}$. The distance between x and any element other than $\bar{x}$ is preserved by induction. Take a neighbor y of x, and consider the lattice $L_2(k-1)$ that consists of the elements that are on a shortest path between y and $\bar{x}$ in the Hasse diagram of $L_2(k)$. Using induction, the distance of $f(y)$ and $f(\bar{x})$ is $k-1$, and all the neighbors of $f(y)$ that are at distance $k-2$ from $f(\bar{x})$ are in $f(L_2(k-1))$. This implies that $f(x)$ must be at distance k from $f(\bar{x})$. □

Corollary 12 *Any weak embedding of $L_2(k)$ to $L_2(n)$ is also a strong embedding.*

Corollary 13 *For any two elements $p, q \in B_n$ that are on the same level or on consecutive levels at distance k, there is a unique embedding of $L_2(k)$ to B_n whose image contains both p and q.*

We will denote the above unique embedding of $L_2(k)$ to B_n by $L_2[p;q]$. (Sometimes we will also use $L_2[p;q]$ to denote the reversal of this poset but this will not lead to confusion, as we will use it only to build two-level posets.)

*The interested reader can find a collection of similar reductions in Booth and Colbourn [1].

2.3 NP-complete 3-uniform hypergraph coloring problems

We will use the **NP**-completeness of MON-NAE-3-SAT, which is (equivalent to) the problem of deciding whether the vertices of a 3-uniform hypergraph are properly 2-colorable, and 3-RAINBOW, which is the problem of deciding whether the vertices of a 3-uniform hypergraph are 3-colorable, such that every hyperedge contains each color exactly once (such colorings are called *rainbow*). The **NP**-completeness of MON-NAE-3-SAT is well-known (it also follows from Schaefer's dichotomy theorem [16]), but we could not find our 3-RAINBOW problem in the literature; it is an easy exercise to show that is **NP**-complete. For completeness, we sketch a proof independently discovered by Jukka Suomela and Antoine Amarilli.*

Proof:(Suomela; Amarilli) Construct a 3-uniform hypergraph $\mathcal{H}$ from a graph G as follows. The vertices of $\mathcal{H}$ are the vertices and edges of G, and the edges of $\mathcal{H}$ are the triples $\{(u, v, uv) \mid uv \text{ is an edge of } G\}$. It is straightforward to see that $\mathcal{H}$ has a rainbow 3-coloring if and only if G has a proper 3-coloring. □

3 Proof of Theorem 1

This section contains the proof of Theorem 1. The problem is trivially contained in **NP**, thus it is enough to prove that it is **NP**-complete to decide whether $P \subset B_{h(P)}$ for an input poset P that has a smallest and a largest element. The reduction is from MON-NAE-3-SAT, the problem of deciding whether the vertices of a 3-uniform hypergraph $\mathcal{H}$ are properly 2-colorable.

The vertices of $\mathcal{H}$ will be denoted by $v_1, \dots, v_n$. The height of the poset P will be $3n$ and we describe its elements over a base set of size $3n$, whose elements we denote by $X = \{a_1, b_1, c_1, a_2, b_2, \dots, c_n\}$. Some elements of P will be defined as subsets of X, with the containment relations preserved, while other elements of P will be defined by their relations to certain subsets of X. The question will be to decide whether P embeds to B_{3n} or not.

P contains every subset of X with at most 9 elements, except the pairs of the form $\{a_i, b_i\}$ and $\{a_i, c_i\}$, and except the sextuples that are *not* of the form $\{a_i, b_i, a_j, b_j, a_k, b_k\}$ or $\{a_i, c_i, a_j, c_j, a_k, c_k\}$. (So P contains $2\binom{n}{3}$ sextuples.) P also contains a chain of length $3n - 8$ for every nonuple S starting at S and ending in X, guaranteeing that S has to be at least $3n - 9$ levels lower than X. (This requires at most $\binom{n}{9}(3n - 10)$ additional elements.) Thus, the smallest element of P is the empty set, and its largest element will be X. This implies that if $P \subset B_{3n}$, then all the elements of P defined so far really must be on the same level as the subset of X that was used to define them. Moreoever, after a suitable renaming/permutation of the base set, it can even be achieved that they are all mapped to exactly to the set defining them.

Now we describe the elements of P that depend on the hypergraph $\mathcal{H}$. These are not defined as a subset of X but by their relations to some of the earlier defined subsets.

P contains for each vertex v_i an element denoted by x_i such that $\{a_i\} < x_i < \{a_i, b_i, c_i\}$. Thus if $P \subset B_{3n}$, then $x_i = \{a_i, b_i\}$ or $\{a_i, c_i\}$.

Finally, P contains for every hyperedge $y_\ell = \{v_i, v_j, v_k\}$ an element Z_ℓ for which $x_i, x_j, x_k < Z_\ell < \{a_i, b_i, c_i, a_j, b_j, c_j, a_k, b_k, c_k\}$. Thus if $P \subset B_{3n}$, then Z_ℓ has to be the unique sextuple that is above x_i, x_j, x_k, so its position is determined by the choice of x_i, x_j, x_k.

As $\{a_i, b_i, a_j, b_j, a_k, b_k\}$ and $\{a_i, c_i, a_j, c_j, a_k, c_k\}$ must have the respective elements of P mapped onto them in any weak embedding of P to B_{3n}, we have $P \subset B_{3n}$ if and only if there is a choice of the position of the elements x_i such that for no hyperedge $\{v_i, v_j, v_k\}$ we have ($x_i = \{a_i, b_i\}$ and $x_j = \{a_j, b_j\}$ and $x_k = \{a_k, b_k\}$) or ($x_i = \{a_i, c_i\}$ and $x_j = \{a_j, c_j\}$ and $x_k = \{a_k, c_k\}$). But if $x_i = \{a_i, b_i\}$ corresponds to coloring v_i red and $x_i = \{a_i, c_i\}$ corresponds to coloring v_i blue, this is clearly equivalent to whether $\mathcal{H}$

*See `http://cstheory.stackexchange.com/a/353/419` and `http://cstheory.stackexchange.com/a/36002/419`.

is 2-colorable or not.

This finishes the proof of Theorem 1.

4 Proof of Theorem 5

This section contains the proof of Theorem 5. The problem is trivially contained in **NP**, thus it is enough to prove that it is **NP**-complete to decide whether a given poset P has a weak embedding to the union of the third and fourth levels of some Boolean lattice. The reduction is from 3-RAINBOW, which is the problem of deciding whether the vertices of a 3-uniform hypergraph have a rainbow 3-coloring, i.e., a 3-coloring where every hyperedge contains each color exactly once.

Now we describe the elements of the two-level poset P that we construct from $\mathcal{H}$. Most elements of P will be defined by subsets of an unspecified base set, with the containment relations preserved.

There is an element $\{a, b, c\}$ that can be thought of as the center of P, and will be the (unique) element with the most neighbors among all elements of P. In any embedding $\{a, b, c\}$ will have to go somewhere on the third level, as there are several elements that are bigger than it, thus, with a slight abuse of notation, we can suppose that it goes to $\{a, b, c\}$.

For every vertex v_i, add an element $\{a, b, c, x_i\}$ to P, and for every hyperedge y_ℓ, add an element $\{a, b, c, z_\ell\}$ to P (where x_i and z_ℓ are different for each vertex and for each hyperedge). We can again suppose that these elements are mapped to "themselves". The way the elements corresponding to vertices and hyperedges can be distinguished is that each $\{a, b, c, x_i\}$ has only one other neighbor, COL_i, which thus can be mapped to either $\{a, b, x_i\}$, $\{a, c, x_i\}$ or $\{b, c, x_i\}$, but each $\{a, b, c, z_\ell\}$ has three further neighbors, $Z_{\ell,i}$, $Z_{\ell,j}$ and $Z_{\ell,k}$, where $y_\ell = \{v_i, v_j, v_k\}$. The three neighbors, $Z_{\ell,i}$, $Z_{\ell,j}$ and $Z_{\ell,k}$, need to be mapped in some permutation to the three neighbors of $\{a, b, c, z_\ell\}$ that are different from $\{a, b, c\}$, i.e., to $\{a, b, z_\ell\}$, $\{a, c, z_\ell\}$ and $\{b, c, z_\ell\}$.

Finally, for every vertex $v_i \in y_\ell$, there is an element $X_{i,\ell}$ that has two neighbors, COL_i and $Z_{\ell,i}$. Therefore, $X_{i,\ell}$ and $Z_{\ell,i}$ must be mapped either to $\{a, b, x_i, z_\ell\}$ and $\{a, b, z_\ell\}$, or to $\{a, c, x_i, z_\ell\}$ and $\{a, c, z_\ell\}$, or to $\{b, c, x_i, z_\ell\}$ and $\{b, c, z_\ell\}$, depending on COL_i.

We now have to show that P can be weakly embedded to the union of the third and fourth levels of some B_n if and only if $\mathcal{H}$ has a rainbow 3-coloring. If $\mathcal{H}$ has a rainbow 3-coloring, then let the image of COL_i be $\{a, b, x_i\}$ if v_i is colored with the first color, $\{a, c, x_i\}$ if v_i is colored with the second color, and $\{b, c, x_i\}$ if v_i is colored with the third color. From this the embedding of $X_{i,\ell}$ and $Z_{\ell,i}$ follows. The fact that all three colors appear at each hyperedge $y_\ell = \{v_i, v_j, v_k\}$ guarantees that the three neighbors of $\{a, b, c, z_\ell\}$, $Z_{\ell,i}$, $Z_{\ell,j}$ and $Z_{\ell,k}$, will not conflict with each other. If P has an embedding, then a rainbow 3-coloring of $\mathcal{H}$ can be derived in a similar way.

This finishes the proof of Theorem 5.

Remark 14 *The above constructed poset P can in fact be embedded to the union of the χ-th and $(\chi+1)$-st levels of some B_n if and only if $\mathcal{H}$ has a rainbow χ-coloring. To see this, the above proof needs to be modified only in that $\{a, b, c\}$ has to go to some set with χ elements, and thus there are χ choices instead of three for the image of each COL_x.*

5 Proof of Theorem 3

This section contains the proof of Theorem 3. The main idea is similar to the proof of Theorem 5, but it is more complicated, and we extensively use Observation 11. As before, the **NP**-membership is trivial, and we prove **NP**-hardness by constructing a poset P from a hypergraph $\mathcal{H}$ such that $\mathcal{H}$ has a rainbow 3-coloring if and only if $e(P) \leq 1$, i.e., if P can be embedded to some two consecutive levels of a Boolean

lattice. We will denote the union of "these" two levels by L_2. This is a bit of a cheating, since we do not know which two levels of which Boolean lattice P could be embedded to. One can think of L_2 either as the union of two sufficiently large levels, or even as the union of two infinite levels, for which our question could be equivalently formulated.

Now we describe the elements of the two-level poset P. Most elements of P will be defined by subsets of an unspecified base set, with the containment relations preserved.

There will be two elements, $\{a, b, c\}$ and $\{p, q, r\}$, which play a central role in the construction. P will contain all $\binom{6}{3} + \binom{6}{4}$ elements of $L_2[\{a, b, c\}; \{p, q, r\}]$. Observation 11 implies that when we weakly embed P to L_2, then the distance of the images of $\{a, b, c\}$ and $\{p, q, r\}$ will be six, thus we can conclude that a, b, c, p, q and r must all be different. We can also suppose that $\{a, b, c\}$ and $\{p, q, r\}$ are, respectively, mapped to some elements $\{a, b, c, W\}$ and $\{p, q, r, W\}$ (which we can consider as "themselves") where W contains some additional elements of the base set.

For every hyperedge y_ℓ, we add $L_2[\{a, b, c, z_\ell\}; \{p, q, r, z_\ell\}]$ to P (where z_ℓ is different for each hyperedge). With another application of Observation 11, we can suppose that these elements are mapped to "themselves + W".

For every vertex v_i, we add two neighboring vertices, $\{a, b, c, x_i\}$ and COL_i to P. We can suppose that $\{a, b, c, x_i\}$ is mapped to $\{a, b, c, x_i, W\}$. COL_i is ideally mapped to one of $\{a, b, x_i, W\}$, $\{a, c, x_i, W\}$ and $\{b, c, x_i, W\}$; for this, we have to eliminate the possibility of it being mapped to some $\{a, b, c, x_i, W \setminus \{w\}\}$. This is why we needed all the complications compared to the construction used to prove Theorem 5.

Finally, for every vertex x_i that is in the hyperedge z_ℓ, we add one more degree two element, $X_{i,\ell}$, that is connected to COL_i and $Z_{\ell,i}$. The element $Z_{\ell,i}$ will be one of the elements from $L_2[\{a, b, c, z_\ell\}; \{p, q, r, z_\ell\}]$ that neighbors $\{a, b, c, z_\ell\}$, i.e., one of $\{a, b, z_\ell\}$, $\{a, c, z_\ell\}$ and $\{b, c, z_\ell\}$. Using Observation 11, we know that $Z_{\ell,i}$ has to be embedded as one of $\{a, b, z_\ell, W\}$, $\{a, c, z_\ell, W\}$ and $\{b, c, z_\ell, W\}$. Therefore, $X_{i,\ell}$ must be mapped either to $\{a, b, x_i, z_\ell, W\}$, $\{a, c, x_i, z_\ell, W\}$ or $\{b, c, x_i, z_\ell, W\}$, and thus COL_i to $\{a, b, x_i, W\}$, $\{a, c, x_i, W\}$ or $\{b, c, x_i, W\}$.

We now have to show that $\mathcal{H}$ has a rainbow 3-coloring if and only if P can be weakly embedded to L_2. If $\mathcal{H}$ has a rainbow 3-coloring, then let the image of COL_i be $\{a, b, x_i, W\}$ if v_i is colored with the first color, $\{a, c, x_i, W\}$ if v_i is colored with the second color, and $\{b, c, x_i, W\}$ if v_i is colored with the third color. From this the embedding of $X_{i,\ell}$ and $Z_{\ell,i}$ follows. The fact that all three colors appear at each hyperedge $y_\ell = \{v_i, v_j, v_k\}$ guarantees that the three neighbors of $\{a, b, c, z_\ell, W\}$, $Z_{\ell,i}$, $Z_{\ell,j}$ and $Z_{\ell,k}$, will not conflict with each other. If P has an embedding, then a rainbow 3-coloring of $\mathcal{H}$ can be derived in a similar way.

This finishes the proof of Theorem 3.

6 Proof of Theorem 7

The vertices of the Johnson graph $J(n, k)$ are the k-element subsets of an n-element base set, and two vertices are connected if they differ in exactly two elements. A graph G is an *induced Johnson subgraph* if there exists an induced copy of G in $J(n, k)$ for some n, k. These graphs were defined in [14] and later studied in [12]. The rest of this section contains a sketch of the proof of Theorem 7. (The details are omitted due to the similarity to the proof of Theorem 3.)

The problem is trivially in **NP**. We prove **NP**-hardness by constructing a graph G from any 3-uniform $\mathcal{H}$ such that G is an induced Johnson subgraph if and only if $\mathcal{H}$ has a rainbow 3-coloring. We need the following variant of Observation 11, which can be similarly proved by induction.

Observation 15 *For any n, k, n', k', any embedding of $J(n, k)$ to $J(n', k')$ is distance-preserving.*

Denote the vertices of $\mathcal{H}$ by $v_1, \dots, v_n$ and its hyperedges by $y_1, \dots, y_m$. Now we describe how to construct G from $\mathcal{H}$.

G will contain a clique on $n+m$ vertices, $x_1, \dots, x_n, z_1, \dots, z_m$ (to be mapped to $\{a, b, c, x_i\}$ and $\{a, b, c, z_\ell\}$), and another clique on m vertices, $z'_1, \dots, z'_m$ (to be mapped to $\{p, q, r, z_\ell\}$).

G also contains a disjoint copy of $J(6,3)$ (which is the same as the edge graph of a cube) for each pair z_ℓ, z'_ℓ, such that z_ℓ and z'_ℓ are contained in this copy of $J(6,3)$ at distance three from each other. These embeddings are unique due to Observation 15.

Finally, G contains a vertex $XZ_{i,\ell}$ (to be mapped to either $\{a, b, x_i, z_\ell\}$, $\{a, c, x_i, z_\ell\}$, or $\{b, c, x_i, z_\ell\}$, depending on the color of v_i) for each $v_i \in y_\ell$. $XZ_{i,\ell}$ is connected to x_i, z_ℓ, and each other vertex of the form $XZ_{i,\ell'}$. (Thus the vertices $(x_i, XZ_{i,\ell}, XZ_{i,\ell'}, \dots)$ form a clique whose size is one more than the degree of v_i in $\mathcal{H}$.)

Similarly to the proof of Theorem 3, it can be proved that the only possible embedding of G to a Johnson graph is the one described in the construction (with a possible extra W in each set). The fact that $XZ_{i,\ell}$ and $XZ_{j,\ell}$ are not neighbors guarantees that every hyperedge must indeed have all three colors.

This finishes the sketch of the proof of Theorem 7.

7 Open problems

We have seen that determining $d(P)$ and $e(P)$ exactly is hard, but is it possible to efficiently approximate these parameters? By placing a copy of P above another copy of P (i.e., all elements of one copy are larger than any element of the other copy), we obtain a poset $P+P$ for which $d(P+P) = 2d(P)+1$ and $e(P+P) = 2e(P)+2$, if P has a smallest and a largest element. This shows that we cannot hope for an additive constant approximation.

On the other hand, by Mirsky's theorem (the dual of Dilworth's theorem), one can partition any poset P on n elements to $h+1 = h(P)+1$ antichains on $n_0, \dots n_h$ elements where $\sum_{i=0}^{h} n_i = n$, and embed these antichains one above the other. For an antichain A_i on n_i elements $d(A_i) \le 1 + \log n_i$, thus $d(P) \le \sum_{i=0}^{h} 1 + \log n_i \le h + h \log \frac{n}{h}$. (It was proved by Grósz, Methuku and Tompkins [9] that almost the same upper bound also holds even for $\pi(P)$. They have also noted that the upper bound is almost sharp if $n_i \approx n/h$ for all i.) From below we trivially have both $\log n \le d(P)$ and $h \le d(P)$, thus this gives a 2-approximation for $\log d(P)$.

It would be interesting to close the gap between these bounds.

Acknowledgements. I would like to thank Balázs Patkós for calling my attention to the problem, and thank him, Balázs Keszegh, Máté Vizer, Abhishek Methuku and Joshua Cooper for discussions. I would also like to thank an anonymous reviewer for several useful suggestions on improving the presentation of the results.

References

[1] K. S. Booth, C. J. Colbourn, Problems polynomially equivalent to graph isomorphism, Report CS-77-04, Computer Science Department, University of Waterloo, 1979. Available at `cs.uwaterloo.ca/research/tr/1977/CS-77-04.pdf`.

[2] B. Bukh, Set families with a forbidden subposet, Electronic J. of Combinatorics, 16 (2009), R142, 11p.

[3] S. A. Choudum, S. Lavanya, Embedding a subclass of trees into hypercubes, Discrete Mathematics 311 (2011), 866-871.

[4] S. A. Cook, The complexity of theorem-proving procedures, Proc. 3rd ACM Symposium on Theory of Computing (1971), 151-158.

[5] J. R. Griggs, 2013, `www.iwoca.org/problems/Griggs.pdf`.

[6] J. R. Griggs, W.-T. Li, Progress on poset-free families of subsets, in Recent Trends in Combinatorics (A. Beveridge, J. R. Griggs, L. Hogben, G. Musiker, P. Tetali, eds), Springer, Berlin, 2016, 317-338.

[7] J.R. Griggs, W.-T. Li, L. Lu, Diamond-free families. J. Comb. Theory (Ser. A) 119 (2012), 310-322.

[8] J.R. Griggs, L. Lu, On families of subsets with a forbidden subposet. Comb. Probab. Comput. 18 (2009), 731-748.

[9] D. Grósz, A. Methuku, C. Tompkins, An Improvement of the General Bound on the Largest Family of Subsets Avoiding a Subposet, to appear in Order.

[10] M. Habib, L. Nourine, O. Raynaud, E. Thierry, Computational aspects of the 2-dimension of partially ordered sets, Theor. Comput. Sci. 312 (2004), 401-431.

[11] M. Livingston, Q. F. Stout, Embeddings in hypercubes, Mathematical and Computer Modelling 11 (1988), 222-227.

[12] M. A. Malik, A. Ali, Some results on induced subgraphs of Johnson graphs, International Mathematical Forum 7 (2012), 445-454.

[13] T. Mütze, Proof of the middle levels conjecture, Proc. Lond. Math. Soc. 112 (2016), 677-713.

[14] R. Naimi, J. Shaw, Induced Subgraphs of Johnson Graphs, `http://arxiv.org/abs/1008.0595`.

[15] O. Raynaud, E. Thierry, The complexity of embedding orders into small products of chains, Order 27 (2010), 365381.

[16] T. J. Schaefer, The complexity of satisfiability problems, in Proceedings of 10th Symposium on Theory of Computing (STOC), ACM Press, New York (1978), 216-226.

[17] J. Stahl, R. Wille, Preconcepts and set representation of contexts, in Classification as a tool of research (W. Gaul, M. Schader, eds.), North-Holland, Amsterdam (1986), 431-438.

[18] W. T. Trotter, Embedding finite posets in cubes, Discrete Math. 12 (1975), 165-172.

Some observations on the traveling salesman problem

Gyula Pap[1]

Egerváry Research Group
Eötvös University
Budapest, Hungary
papgy@cs.elte.hu

Abstract: In this talk we elaborate on an approach to the traveling salesman problem based on linear programing duality - with the main goal in our sight being the conjecture that there is an approximation algorithm that has a bound better than the 3/2 guaranteed by Christofides algorithm, compared against the Held-Karp linear programing relaxation. The key observation is that by considering the problem of fractionally packing tour vectors, we can re-formulate equivalent conjectures in which there is no need to consider a metric space as in the usual formulation of the traveling salesman problem, and the usual discussion of algorithms. Instead, we will rely on the fractional packing of tour vectors. The main objective here is to describe some alternative version of the conjecture, and consider some special cases.

Keywords: traveling salesman problem, approximation algorithm, cubic graph

1 Introduction

Chritofides' algorithm finds a tour in a complete graph with metric edge-weights that is no longer than 3/2 times the shortest one actually, by an analysis given by Wolsey, the tour found will be no longer than 3/2 times the Held-Karp linear programing bound. Let us formulate this more precisely.

Edge-weights in a complete graph are called metric if it non-negative and satisfies the triangle inequality.

Problem 1 (Metric TSP v.1) *Given a complete graph $G = (V, E)$ and metric edge-weights $w : E \to \mathbb{R}_+$, minimize the weight of a Hamiltonian cycle.*

We may consider an equivalent version of the traveling salesman problem by considering tours in arbitrary graphs.

Definition 2 *A* tour *of a graph is a closed walk that visits every node at least once.*

Problem 3 (Metric TSP v.2) *Given a graph $G = (V, E)$ and edge-weights $w : E \to \mathbb{R}_+$, minimize the weight of a tour.*

Equivalence between Problem 1 and Problem 3 is easily seen by shortcutting a tour to find a Hamiltonian cycle.

A well-known lower bound is determined by the Held-Karp linear programming relaxation [4], described as follows, for vectors $x : E \to \mathbb{R}$.

$$\begin{aligned} &\min wx \quad \text{s.t.} \\ &\qquad x \geq 0 \\ &\sum_{uv \in \delta(U)} x(uv) \geq 2 \text{ for all } U \subsetneq V, U \neq \emptyset \end{aligned} \tag{1}$$

[1]Research is supported by the Hungarian Academy of Sciences.

It is quite easy to see that this is a lower bound on the optimum in Problem 3 – basically it requires the tour to cross every cut at least twice. Sometimes the equation $\sum_{u:uv\in E} x(uv) = 2$ for all $v \in V$ is added when looking for a Hamiltonian cycle, though here, as we will be looking for tours, we omit these equations. It is believed that this omission does not lessen our chances of finding a good approximation bound.

Theorem 4 (Christofides' algorithm [2], Wolsey's analysis [6]) *A tour found by Christofides' algorithm has weight no more than 3/2 times the Held-Karp LP bound in* (1).

Corollary 5 *The Held-Karp LP has integrality gap no more than 3/2.*

2 Combination of tour-vertors

The main plot is to turn the statement in Corollary 5 "upside down" by linear programing duality – for this we need the notion of a tour-vector. Note that a tour-vector is equal to the edge-frequency vector for some tour, and it is straightforward to find a tour like that.

Definition 6 *A vector $t : E \to \{0, 1, 2\}$ is called a tour-vector if its support is the edge-set of a connected graph on the node set V, and for all nodes v the sum $\sum_{uv\in E} x(uv)$ is even.*

Thus we can formulate the following theorem, that is (see below) derived from Theorem 4 by duality, in a way.

Theorem 7 *Suppose we are given a graph $G = (V, E)$, and a vector $x' : E \to \mathbb{R}_+$ that satisfies*

$$\begin{gathered} x' \geq 0 \\ \sum_{uv\in\delta(U)} x'(uv) \geq 3 \quad \text{for all } U \subsetneq V, U \neq \emptyset \end{gathered} \tag{2}$$

Then there is a convex combination of tours, that is a number $k \in \mathbb{Z}_+$, tour-vectors $t_1, t_2, \cdots, t_k$, and k real numbers $\lambda_1, \lambda_2, \cdots, \lambda_k \in \mathbb{R}$ such that $\lambda_i \geq 0$ for all i and $\sum_{i=1}^k \lambda_i = 1$ and satisfying

$$\sum_{i=1}^{k} \lambda_i t_i(uv) \leq x'(uv) \quad \text{for all } uv \in E. \tag{3}$$

Here (by Carateodory's Theorem) we may assume $k \leq |E| + 1$.

Proof: First, let $\mathcal{T}(G)$ denote the set of tour vectors in G, and then we define the "up hull" of tour vectors as follows:

$$up.hull(\mathcal{T}(G)) := conv.hull(\mathcal{T}(G)) + \mathbb{R}_+^E :=$$

$$= \{u : \text{ there is } k \in \mathbb{Z}_+, \text{ tour-vectors } t_1, t_2, \cdots, t_k, \text{ real numbers } \lambda_1, \lambda_2, \cdots, \lambda_k \in \mathbb{R}$$
$$\text{such that } \lambda_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{k} \lambda_i = 1 \text{ and satisfying } u \geq \sum_{i=1}^{k} \lambda_i t_i(uv) \leq x'(uv) \}$$

Note that $up.hull(\mathcal{T}(G))$ is a polyhedron that is "closed upwards", i.e. adding a non-negative vector to any member of this polyhedron it stays in the polyhedron. This implies the following claim.

Claim 8 *Any inequality $ax \geq b$ valid for $up.hull(\mathcal{T}(G))$ has only non-negative coefficients, that is $a \geq 0$.*

Now getting back to proving Theorem 7, by contradiction let us assume that there is no convex combination as required, that is, assume that $x' \notin up.hull(\mathcal{T}(G))$. Let $ax \geq b$ denote a separating inequality that is,

$$ax' < \min_{x \in up.hull(\mathcal{T}(G))} ax'. \tag{4}$$

Then consider Problem 3 for graph $G = (V, E)$ and cost vector $w := a$. Since $\frac{2}{3}x'$ satisfies the Held-Karp LP inequalities, the Held-Karp optimum is less than or equial to $\frac{2}{3}ax'$. By Theorem 4, Christofides' algorithm finds a tour (vector) that has weight less than or equal to $\frac{3}{2}$ times the Held-Karp optimmum, that is, a tour vector of weight less than or equal to ax'. This contradicts (4), which proves Theorem 7. □

Thought we know this theorem as it follows from Theorem 4, it seems far from obvious how to *construct* efficiently such a convex combination. One way is based on the ellipsoid method, the equivalence of separation amd optimization, and uses Christofides' algorithm as a tool for separation (suggested by A. Jüttner). This approach has various technical issues regarding details of the ellipsoid method, but it plausibly implies a polynomial time algorithm to find such a convex combination. However, it would be desirable to have a more direct, more combinatorial approach to do just this – such and approach may lead to new results, and may help in proving Conjecture 7 (see below) for a value $\alpha < 3/2$.

The connection between Theorem 4 and this Theorem 7 goes both ways. Theorem 7 is derived from 4 by an LP duality argument. Vice versa, given a graph G and weights w, we can solve Theorem 4 by applying Theorem 7 as follows: solve the Held-Karp LP to find x, and then apply Theorem 7 for $x' := 3/2x$ – which satisfies (2) – to find the tours $t_1, t_2, \cdots, t_k$. Then we minimize wt_i to find a tour $t := t_i$, and obtain a tour that has weight no more than 3/2 times the Held-Karp optimum. This is a "best of many" type approach – this term has been coined by a similar approach of Ahn, Kleinberg and Shmoys [1] for the TSP-path problem.

3 Observations

An "upside down" way to describe the 3/2 approximation obtained by Christofides' algorithm is given in Theorem 7 – and it is quite natural to formulate the conjecture that, if the "3" is replaced by a smaller value in (2), then we obtain a better approximation. This gives the following "α-conjecture" that, where "3" is replaced by 2α the point is that if "α"-conjecture holds for a given value of α, then that implies an α-approximation result (see below).

For $\alpha = 3/2$, this conjecture is true, we just get back Theorem 7. It is false for values $\alpha < 4/3$. For any value $4/3 \leq \alpha < 3/2$, this conjecture is open.

Conjecture 9 ("α" combinations) *Suppose we are given in a graph $G = (V, E)$, and a vector $x' : E \to \mathbb{R}_+$ that satisfies*

$$\begin{aligned} x' &\geq 0 \\ \sum_{uv \in \delta(U)} x'(uv) &\geq 2\alpha \quad \textit{for all } U \subsetneq V, U \neq \emptyset \end{aligned} \tag{5}$$

Then there is a convex combination of tours, that is a number $k \in \mathbb{Z}_+$, tour-vectors $t_1, t_2, \cdots, t_k$, and k real numbers $\lambda_1, \lambda_2, \cdots, \lambda_k \in \mathbb{R}$ such that $\lambda_i \geq 0$ for all i and $\sum_{i=1}^{k} \lambda_i = 1$ and satisfying

$$\sum_{i=1}^{k} \lambda_i t_i(uv) \leq x'(uv) \quad \textit{for all } uv \in E. \tag{6}$$

If for a value of $\alpha \geq 1$ Conjecture 9 holds, then that implies the followiong conjecture for the same value of α.

Conjecture 10 ("α" gap) *The Held-Karp LP has integrality gap no more than α.*

If for a value of $\alpha \geq 1$ Conjecture 9 holds, then – assuming that there is a efficient way to *find* the convex combination – that implies the followiong conjecture for the same value of α.

Conjecture 11 ("α" approximation) *There is an α-approximation algorithm for metric TSP.*

4 A special case

A special case of Theorem 7 is when x' is constant 1 for all edges of the graph – thus we obtain the following theorem.

Theorem 12 *Let $G = (V, E)$ be 3-regular 3-connected graph. Then there is a convex combination of tour vectors, that is there is a number $k \in \mathbb{Z}_+$, and k real numbers $\lambda_1, \lambda_2, \cdots, \lambda_k \in \mathbb{R}$ such that $\lambda_i \geq 0$ for all i and $\sum_{i=1}^{k} \lambda_i = 1$ and satisfying*

$$\sum_{i=1}^{k} \lambda_i t_i(uv) \leq 1 \quad \textit{for all } uv \in E. \tag{7}$$

This is a special case of Theorem 7, so a proof based on linear programming and an algorithm based on the ellipsoid algorithm we already have as shown above. Below we will show a different, more combinatorial approach. The following approach is based on discussions with T. Király. The motivation in cosidering this approach is to try to prove a special case of Conjecture 9 for some $\alpha < 3$ – in this case, this boils down to proving the strengthening of this Theorem 12 when we replace "≤ 1" by "≤ 0.99" in (7). Unfortunately, this effort has proved unsuccessful as of the time of writing of this extended abstract. Typical counterexamples show that the best value in the right hand side of (7) we can hope for, in place of 0.99 is $8/9 \approx 0.888$ – that would imply a 4/3-approximation. Anyway, we believe that the following proof bears some relevance in future efforts in new TSP-related results.

The second, more combinatorial proof of Theorem 12 is based on the following Lemma.

Lemma 13 *If $G = (V, E)$ is a 3-regular 3-connected graph, then there is a perfect matching $M \subseteq E$ for which there is no 3-edge cut that is contained in M.*

Proof: If the graph is 4-connected, then any perfect matching (which exists by Petersen's theorem) will do. Otherwise consider an inclusionwise minimal set U such that $U \subseteq V$, $|U| > 1$, $\delta(U) = 3$. Then G/U is 3-connected and 3-regular, thus by induction, a perfect matching M_1 exists in G/U. Further, $G/(V-U)$ is 4-edge-connected and 3-regular. Let $e \in M_1 \cap \delta(U, V-U)$ be the edge of M_1 in the cut. Then there is a perfect matching M_2 in $G/(V-U)$ such that $e \in M_2$. $M := M_1 \cup M_2$ is a perfect matching in G as required. □

Proof: (of Theorem 12) By Lemma 13, let M be a perfect matching that does not contain all 3 edges of any 3-edges cut. Then $E-M$ is a set of vertex disjoint cycles $C_1, C_2, \cdots C_m$. Let G' denote the graph of m nodes obtained from G by shrinking $C_1, C_2, \cdots, C_m$. Because of our choice of M, G' is 4-edge-connected, and thus there are 2 disjoint spanning trees in G', say T_1 and T_2. Let t_1 and t_2 denote the tour vectors obtained such that $t_i(uv) := 1$ for $uv \in \cup_i C_i$ and $t_i(uv) := 2$ for $uv \in T_i$. Define $\lambda_1 := \lambda_2 := 1/2$. Then $t_1, t_2, \lambda_1, \lambda_2$ are a convex combination as required in Theorem 12. □

5 Acknowledgment

The author is grateful to András Sebő, Attila Bernáth, Tamás Király, Kristóf Bérczi, Alpár Jüttner, Zoltán Király, and Csaba Király for discussions on the topic.

References

[1] H.-C. An, R. Kleinberg, and D. B. Shmoys, Improving Christofides' algorithm for the s-t path TSP, Journal of the ACM, 62(5), 2015.

[2] N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem, Report 388, Graduate School of Industrial Administration, CMU, 1976.

[3] I. Czeller, G. Pap, A note on bounded weighted graphic metric TSP Egerváry Research Group Technical Report TR-2014-03

[4] M. Held, R.M. Karp, A dynamic programming approach to sequencing problems, Journal for the Society for Industrial and Applied Mathematics 1:10. 1962

[5] A. Sebő, J. Vygen, Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs, Combinatorica (2014)

[6] L. A. Wolsey, Heuristic analysis, linear programming and branch and bound. Combinatorial Optimization II, volume 13 of Mathematical Programming Studies, 121134. Springer Berlin Heidelberg, 1980

Forbidden Pairs of Minimal Quadratic and Cubic Configurations

Attila Sali[1]

Alfréd Rényi Institute of Mathematics
Hungarian Academy of Sciences
sali.attila@renyi.mta.hu

Sam Spiro[2]

University of Miami
sam.a.spiro@gmail.com

Abstract: A matrix is *simple* if it is a (0,1)-matrix and there are no repeated columns. Given a (0,1)-matrix F, we say a matrix A has F as a *configuration*, denoted $F \prec A$, if there is a submatrix of A which is a row and column permutation of F. Let $|A|$ denote the number of columns of A. Let $\mathcal{F}$ be a family of matrices. We define the extremal function $\text{forb}(m, \mathcal{F}) = \max\{|A|\colon A \text{ is an } m - \text{rowed simple matrix and has no configuration } F \in \mathcal{F}\}$. We consider pairs $\mathcal{F} = \{F_1, F_2\}$ such that F_1 and F_2 have no common extremal construction and derive that individually each $\text{forb}(m, F_i)$ has greater asymptotic growth than $\text{forb}(m, \mathcal{F})$, extending research started by Anstee and Koch [7].

Hypergraph trace, forbidden configuration, extremal graph theory

1 Introduction

The investigations into the extremal problem of the maximum number of edges in an n vertex graph with no subgraph H originated with Erdős and Stone [13] and Erdős and Simonovits [12] . There is a large and illustrious literature. A natural extension to general hypergraphs is to forbid a given *trace*. This latter problem in the language of matrices is our focus. We say a matrix is *simple* if it is a (0,1)-matrix and there are no repeated columns. Given a (0,1)-matrix F, we say a matrix A has F as a *configuration*, denoted $F \prec A$, if there is a submatrix of A which is a row and column permutation of F. Let $|A|$ denote the number of columns in A. We define

$$\text{Avoid}(m, F) = \{A : A \text{ is } m\text{-rowed simple}, F \not\prec A\},$$

$$\text{forb}(m, F) = \max_A\{|A| : A \in \text{Avoid}(m, F)\}.$$

A simple (0,1)-matrix A can be considered as vertex-edge incidence matrix of a hypergraph without repeated edges. A configuration is a trace of a subhypergraph of this hypergraph.

Let A^c denote the 0-1-complement of a (0,1)-matrix A. It is easy to see that $\text{forb}(m, F) = \text{forb}(m, F^c)$.

We recall an important conjecture from [10]. Let I_k denote the $k \times k$ identity matrix, let I_k^c denote the (0,1)-complement of I_k, and let T_k denote the $k \times k$ upper triangular matrix whose ith column has 1's in rows $1, 2, \ldots, i$ and 0's in the remaining rows. For p matrices $m_1 \times n_1$ matrix A_1, an $m_2 \times n_2$ matrix A_2,..., an $m_p \times n_p$ matrix A_p we define $A_1 \times A_2 \times \cdots \times A_p$ as the $(m_1 + \cdots + m_p) \times n_1 n_2 \cdots n_p$ matrix whose columns consist of all possible combinations obtained from placing a column of A_1 on top of a column of A_2 on top of a column of A_3 etc. For example, the vertex-edge incidence matrix of the complete bipartite graph $K_{m/2,m/2}$ is $I_{m/2} \times I_{m/2}$. Define 1_k to be the $k \times 1$ column of 1's and 0_ℓ to be the $\ell \times 1$ column of 0's.

[1]Research was partially supported by Hungarian National Research, Development and Innovation Office - NKFIH, K116769

[2]Research was done while the second author took the *Research Opportunties* course at Budapest Semesters in Mathematics under the supervision of the first author

Conjecture 1 *[10] Let F be a $k \times \ell$ matrix with $F \neq \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Let $X(F)$ denote the largest p such that there are choices $A_1, A_2, \ldots, A_p \in \{I_{m/p}, I^c_{m/p}, T_{m/p}\}$ so that $F \not\prec A_1 \times A_2 \times \cdots \times A_p$. Then* $\mathrm{forb}(m, F) = \Theta(m^{X(F)})$.

We are assuming p divides m which does not affect asymptotic bounds.

It is natural to extend the concepts of $\mathrm{Avoid}(m, F)$ and $\mathrm{forb}(m, F)$ to the case when not just a single configuration, but a family $\mathcal{F} = \{F_1, F_2, \ldots, F_r\}$ of configurations is forbidden.

$$\mathrm{Avoid}(m, \mathcal{F}) = \{A : A \text{ is } m\text{-rowed simple}, F \not\prec A \text{ for all } F \in \mathcal{F}\},$$

$$\mathrm{forb}(m, \mathcal{F}) = \max_A\{|A| : A \in \mathrm{Avoid}(m, \mathcal{F})\}.$$

One important result in this area is the following theorem of Balogh and Bollobás [11].

Theorem 2 (Balogh and Bollobás, 2005) *For a given k, there is a constant $BB(k)$ such that*

$$\mathrm{forb}(m, \{I_k, T_k, I^c_k\}) = BB(k).$$

The best current estimate for $BB(k)$ is due to Anstee and Lu [8], $BB(k) \leq 2^{ck^2}$ where c is absolute constant, independent of k. It could be tempting to extend Conjecture 1 to the case of forbidden families, as well. However, as it was shown in [5] $\mathrm{forb}(m, \{I_2 \times I_2, T_2 \times T_2\})$ is $\Theta(m^{3/2})$ despite the only products missing both $I_2 \times I_2$ and $T_2 \times T_2$.are one-fold products. An even stronger observation is made in Remark 27.

In the present paper we continue the investigations started in [7]. Anstee and Koch determined $\mathrm{forb}(m, \{F, G\})$ for all pairs $\{F, G\}$, where both members are *minimal quadratics*, that is both $\mathrm{forb}(m, F) = \Theta(m^2)$ and $\mathrm{forb}(m, G) = \Theta(m^2)$, but no proper subconfiguration of F or G is quadratic. We take this one step further. That is, we consider cases when one of F or G is a simple minimal cubic configuration and the other one is a minimal quadratic or minimal simple cubic. Our results are summarized in Table 3. We solve all cases when the minimal simple cubic configuration has four rows.

The structure of the paper is as follows. Since volume restrictions do not allow detailed treatment of all cases, we just sample some of the interesting ones. Complete study is to be published in a forthcoming problem. In Section 2 a stability theorem is proven for matrices avoiding the configuration $Q_3(t)$, which is a generalization of the configuration Q_3 (see Table 1), and this theorem is applied to prove forbidden pairs results involving $Q_3(t)$. Section 3 contains cases when one member of the forbidden pairs is a block of 1's. This naturally involves extremal graph and hypergraph results, as forbidding $1_{k,1}$ restricts the hypergraph corresponding to our simple (0,1)-matrix to be of *rank*-$(k-1)$, that is edges are of size at most $k-1$. Interestingly enough, in one case we use a very recent theorem of Alon and Shikhelman [1] combined with an old fundamental result of Füredi [14].

Throughout the paper we use standard extremal graph and hypergraph notations, such as $ex(m, G)$ to denote the largest number of edges a graph on m vertices can have without containing a subgraph isomorphic to G, or $ex^{(k)}(m, \mathcal{H})$ for the largest number of edges a k-uniform hypergraph can have without containing a subhypergraph $\mathcal{H}$. The complete k-partite k-uniform hypergraph on partite sets of sizes $s_1, \ldots, s_k$, respectively is denoted by $K(s_1, \ldots, s_k)$. Also, when forbidden pairs of configurations are considered, we use the notational simplification $\mathrm{forb}(m, \{F, G\}) = \mathrm{forb}(m, F, G)$ for typesetting convenience. We allow ourselves the ambiguity of writing $I \times I^c$ instead of the technically precise $I_{m/2} \times I^c_{m/2}$ in product constructions.

What follows are tables of all minimal quadratic configurations and simple minimal cubic configurations with 4 rows. In addition to the configurations, we have included a list of all 2-fold and 3-fold products of I, I^c and T that avoid these configurations. The list of constructions avoiding quadratic configurations comes from [7].

Note that we have not included the complements of $1_{3,1}$, $1_{2,2}$, and I_3 in this table, even though these are also minimal quadratic configurations. This is because if Q denotes any of these configurations then $\mathrm{forb}(m, Q, F) = \mathrm{forb}(m, Q^c, F^c)$, which is already included in Table 3.

	Configuration Q_i	Construction(s)
$1_{3,1}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	$I \times I$
$1_{2,2}$	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$	$I \times I$
I_3	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$I^c \times I^c$ $I^c \times T$ $T \times T$
Q_3	$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$	$I \times I^c$
Q_8	$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$	$T \times T$
Q_9	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$	$I \times T$ $I^c \times T$

Table 1: Minimal Quadratic Configurations

	Configuration F_i	Quadratic Const.(s)	Cubic Const.(s)
$1_{4,1}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$I \times I$	$I \times I \times I$
F_9	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$	$I^c \times I^c$ $I^c \times T$ $T \times T$	$I^c \times I^c \times T$
F_{10}	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$I^c \times I^c$ $I^c \times T$ $T \times T$	$I^c \times I^c \times T$
F_{11}	$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$	$I \times T$ $I^c \times T$ $T \times T$	$T \times T \times T$
F_{12}	$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	All	All
F_{13}	$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$	All	$T \times T \times T$

Table 2: Minimal Simple Cubic Configurations with 4 Rows

	$1_{4,1}$	F_9	F_{10}	F_{11}	F_{12}	F_{13}	$0_{4,1}$	F_9^c	F_{10}^c	F_{12}^c
$1_{3,1}$	$\Theta(m^2)$	$m+2$	$\Theta(1)$	$\Theta(m^{3/2})$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(1)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$
$1_{2,2}$	$\Theta(m^2)$	$m+3$	$\Theta(1)$	$\Theta(m^{3/2})$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(1)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$
I_3	$\Theta(1)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$
Q_3	$\Theta(m)$	$\Theta(m)$	$\Theta(m)$	$\Theta(m^{3/2})$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m)$	$\Theta(m)$	$\Theta(m)$	$\Theta(m^2)$
Q_8	$\Theta(m)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$
Q_9	$3m-2$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$3m-2$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$
$1_{4,1}$		$m+5$	$\Theta(1)$	$\Theta(m^{3/2})$	$\Theta(m^3)$	$\Theta(m^2)$	$\Theta(1)$	$\Theta(m^3)$	$\Theta(m^3)$	$\Theta(m^3)$
F_9			$\Theta(m^3)$	$\Theta(m^2)$	$\Theta(m^3)$	$\Theta(m^2)$	$\Theta(m^3)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^3)$
F_{10}				$\Theta(m^2)$	$\Theta(m^3)$	$\Theta(m^2)$	$\Theta(m^3)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^3)$
F_{11}					$\Theta(m^3)$	$\Theta(m^3)$	$\Theta(m^{3/2})$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^3)$
F_{12}						$\Theta(m^3)$	$\Theta(m^3)$	$\Theta(m^3)$	$\Theta(m^3)$	$\Theta(m^3$
F_{13}							$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^2)$	$\Theta(m^3)$

Table 3: Results

In addition to this, the complement of $1_{4,1}$ (which we denote as $0_{4,1}$), F_9^c, F_{10}^c, and F_{12}^c are minimal simple cubic configurations, and the products avoiding these configurations are the complements of the products avoiding their complements.

Table 3 contains the asymptotic values for all pairings of the configurations mentioned above when at least one of the configurations is cubic. We note that all exact results stated below hold for m sufficiently large.

Many of the results of Table 3 follow from the following simple observation.

Remark 3 *If F and G are both avoided by the same p-fold product construction then* $\text{forb}(m, F, G) = \Omega(m^p)$.

2 Avoiding $Q_3(t)$

We consider a slight generalization of Q_3

$$Q_3(t) = \begin{bmatrix} 0 & \overbrace{1\cdots1}^{\text{t}} & \overbrace{0\cdots0}^{\text{t}} & 1 \\ 0 & 0\cdots0 & 1\cdots1 & 1 \end{bmatrix},$$

where we always assume $t \geq 2$ when we write $Q_3(t)$. We have the following result from [7].

Theorem 4 $\text{forb}(m, Q_3(t), t \cdot I_k) = \text{forb}(m, Q_3(t), t \cdot I_k^c) = \Theta(m)$ *for any fixed k.*

Corollary 5 $\text{forb}(m, Q_3, F) = \Theta(m)$ *for* $F = 1_{4,1},\ F_{10},\ 0_{4,1},\ F_{10}^c$.

Proof: Each of these F is contained in either I_k or I_k^c for sufficiently large k, so Theorem 4 gives the upper bound, and either I_m or I_m^c gives the lower bound. □

Our main result for this section will be a stability theorem which says that large $Q_3(t)$ avoiding matrices "look like" $I_{m/2} \times I_{m/2}^c$, and from this we will be able to prove an upper bound for $\text{forb}(m, Q_3, F_{11})$, and more generally for $\text{forb}(m, Q_3(t), I_r \times I_s)$. We first introduce some terminology for the proof.

We will say that a row r is *sparse* when restricted to a set of columns C if, restricted to C, r has at least one 0 but fewer than t 0's (i.e. r has few 0's but is not identically 1), and we will say that a row r is *dense* when restricted to a set of columns C if r has at least one 1 and at least t 0's within the columns of C (i.e. r has many 0's but is not identically 0). We will say that a column $c \in C$ is *identified* by a sparse row r if r has a 0 in column c.

If A is a matrix and C is a set of columns (not necessarily a subset of the columns of A), then $A \setminus C$ will denote the set of columns in A that are not in C. We define the matrix $Q_3(t;0)$ to be $Q_3(t)$ without its column of 1's. Lastly, we restate Theorem 4 as follows: for any fixed k and t there exists a constant $c_{k,t}$ such that if A is an m-rowed simple matrix with $|A| > c_{k,t}m$ and $Q_3(t) \not\prec A$, then $t \cdot I_k \prec A$.

Theorem 6 *Let $A \in \text{Avoid}(m, Q_3(t))$ with $|A| = \omega(m \log m)$. There exists a set of integers $\{k_1, \ldots, k_y\}$ and a set $A' = \left\{A'_1, \ldots, A'_y\right\}$, of configurations $A'_j \prec A$ such that:*

1. *$k_{j+1} \leq \frac{1}{2}k_j$ for all j, and $y \leq \log m$.*
2. *There exists k_j rows of A such that the columns of A'_j restricted to these rows are columns of I_{k_j}.*
3. *If i is a column of I_{k_j} and C_i^j is the set of columns in A'_j that are an i column in the rows mentioned above, then no row restricted to C_i^j is dense, and every column of C_i^j is identified by some sparse row.*
4. *$|A| = \Theta(\sum |A'_j|)$.*

We first present an outline of the proof before going into the details. We are given a large $Q_3(t)$ avoiding matrix A_0, and as a first step we remove all rows from A_0 that have few 1's (for technical reasons) to get a new matrix A_1. We then find the largest $t \cdot I_k$ in A_1, and our goal is to use this as the I_{k_1} base for A'_1. To do so, we trim A_1 by getting rid of all columns of C_i^1 that are not identified by a sparse row, as well as all rows that are dense restricted to some C_i^1. This gives us A'_1, and we repeat the process on the remaining columns of A_1, A_2 (after again removing rows with few 1's). It turns out that the largest $t \cdot I$ in A_2, I_{k_2}, will satisfy $k_2 \leq \frac{1}{2}k_1$, and thus we can repeat this process at most $\log m$ times. At each step we remove only $O(m)$ columns, so in total only $O(m \log m)$ columns of A_0 were removed. As $|A_0| = \omega(m \log m)$, the columns that remain (those of A') must be asymptotically as large as our original A_0.

Proof: Let $A_0 \in \text{Avoid}(m, Q_3(t))$ with $|A_0| = \omega(m \log m)$. Let R_1 denote the set of rows of A_0 that have fewer than $3t-2$ 1's, and let A_1 denote A_0 with these rows removed. Note that A_1 need not be a simple matrix, but if C_{R_1} denotes the set of columns that have a 1 in some row of R_1, then $A_1 \setminus C_{R_1}$ will be simple. As $|C_{R_1}| \leq (3t-2)m = O(m)$, $|A_1 \setminus C_{R_1}| = \Theta(|A_0|)$. Note that we will be working with the matrix A_1, *not* its simplification $A_1 \setminus C_{R_1}$, in order to use the fact that every row has at least $3t-2$ 1's.

Define k_1 to be the largest integer such that $t \cdot I_{k_1} \prec A_1$. As $|A_1 \setminus C_1| = \omega(m)$, Theorem 4 tells us that we have $t \cdot I_k \prec A_1 \setminus C_1 \prec A_1$ for any fixed k (so in particular we can assume that $k_1 \geq 3$). Rearrange rows so that this $t \cdot I_{k_1}$ appears in the first k_1 rows of A_1.

Note that no column of A_1 can have two 1's in the first k_1 rows. Indeed, any two rows of $t \cdot I_{k_1}$ for $k_1 \geq 3$ induce a $Q_3(t;0)$, and hence if a column had 1's in two of these rows we would have $Q_3(t) \prec A_1$. We can thus partition the columns of A_1 as follows. We will say that a column c belongs to the set C_i^1 for $1 \leq i \leq k_1$ if c has a 1 in row i, and we will say that $c \in C^2$ if c has no 1's in these rows. We will make the additional assumption that the $t \cdot I_{k_1}$ we placed in the first k_1 rows was such that $|C^2|$ is minimal. Note that $|C_i^1| \geq 3t-2$ for all i, as otherwise the ith row would belong to R_1 and hence not be in A_1.

We now examine the rows that are dense in some C_i^1.

Lemma 7 *If a row r restricted to C_i^1 is dense, then restricted to $A_1 \setminus C_i^1$, r has at most $t-1$ 1's or r is identically 1.*

Proof: Assume r is dense restricted to C_i^1, i.e. it has at least t 0's and one 1 restricted to C_i^1. If r had t 1's and a 0 in $A \setminus C_i^1$, then by looking at the ith row, row r, and the relevant columns, we would find a $Q_3(t)$. $\square$

We would like to strengthen the above lemma to say that dense rows are either identically 0 or identically 1 outside of their C_i^1, and to do so we'll have to ignore a small number of columns of A_1. We will say that a column c is "bad" if there exists a row r and integer i such that r is dense restricted to C_i^1, r is not identically 1 in $A \setminus C_i^1$, and c has a 1 in row r. Let $\overline{C^1}$ denote the set of "bad" columns.

Lemma 8 $|\overline{C^1}| = O(m)$.

PROOF: Each dense row r contributes at most $t-1$ columns to $\overline{C^1}$ by Lemma 7, and hence $|\overline{C^1}| \leq (t-1)m = O(m)$. $\square$

We now wish to ignore the dense rows of A_1, as well as any rows of $\bigcup C_i^1$ that are not identified by a sparse row. Rearrange rows so that the bottom ℓ rows of A_1 consist of all rows that when restricted to some C_i^1 are dense. Let $\widehat{C_i^1}$ denote the columns of C_i^1 that are not identified by a sparse row and that are not in C_{R_1} or $\overline{C^1}$. Let $\widehat{A_1}$ denote A_1 restricted to the top k_1 rows, the bottom ℓ rows, and the columns of $\bigcup \widehat{C_i^1}$.

Lemma 9 *$\widehat{A_1}$ is a simple matrix.*

PROOF:

Let $\hat{c}$ and $\hat{d}$ be columns of $\widehat{A_1}$ with corresponding columns c, d in $A_1 \setminus C_{R_1}$ (as no $\widehat{C_i^1}$ columns are in C_{R_1}). If $\hat{c} = \hat{d}$, then clearly we must have $c, d \in C_i^1$ for some i. As $c \neq d$ (because $A_1 \setminus C_{R_1}$ is a simple matrix), we must have c and d differing in some row r above the bottom ℓ rows, say c has a 0 in row r and d has a 1. But this means that r must be sparse (as every row between the top k_1 rows and bottom ℓ rows is either identically 0, identically 1, or sparse), and hence c is identified by a sparse row, contradicting $\hat{c}$ belonging to $\widehat{A_1}$. $\square$

Lemma 10 $|\widehat{A_1}| = O(m)$.

PROOF: By Lemma 7 (and the fact that $\widehat{A_1}$ contains no columns of $\overline{C^1}$), we know that each row r restricted to $\widehat{C_i^1}$ can be one of four types: r can be identically 0 restricted to $A_1 \setminus C_i^1$ (in which case we will say it is a row of $B_{i,0}$), r can be identically 1 restricted to $A_1 \setminus C_i^1$ (in which case we will say it is a row of $B_{i,1}$), or r can itself be either identically 0 or identically 1. We thus have that the matrix B_i formed by restricting $\widehat{A_1}$ to the columns $\widehat{C_i^1}$ and to the rows of $B_{i,0}$ and $B_{i,1}$ is simple with $|\widehat{C_i^1}|$ columns. Let b_i denote the number of rows in B_i.

If $|B_i| > c_{3,t} b_i$, then we must have $t \cdot I_3 \prec B_i$, and hence either $B_{i,0}$ or $B_{i,1}$ must contain a $Q_3(t;0)$. If $B_{i,1}$ contains a $Q_3(t;0)$, then these rows and columns together with any column of $A_1 \setminus C_i^1$ gives a $Q_3(t)$. If $B_{i,0}$ contains a $Q_3(t;0)$, then one can find a $t \cdot I_{k_1+1}$ in A_1. Indeed, in A_1 (note that we are no longer ignoring the columns of $\overline{C^1}$ and C_{R_1}) take the two rows from $B_{i,0}$ that contain a $Q_3(t;0)$, ignore the at most $2t-2$ columns that have 1's in these rows outside of C_i^1, and swap these rows with rows i and k_1+1. After performing these steps, no column of A_1 has two 1's in any of the first k_1+1 rows (since we removed the at most $2t-2$ columns that could pose a problem), rows i and k_1+1 by assumption have at least t 1's, and as every other row had at least $3t-2$ 1's before ignoring the at most $2t-2$ columns, they all still have at least t 1's. Hence we have $t \cdot I_{k_1+1} \prec A_1$, contradicting our definition of k_1. Thus we must have $|B_i| = |\widehat{C_i}| \leq c_{3,t} b_i$, and in total we have

$$|\widehat{A_1}| = \sum |\widehat{C_i^1}| \leq \sum c_t b_i \leq c_t \ell \leq c_t m,$$

proving the statement. $\square$

We now let A_1' be $\bigcup C_i^1$ after removing the columns of $\widehat{A_1}$, C_{R_1}, and $\overline{C^1}$ (which in total are only of size $O(m)$), along with the bottom ℓ rows. If $|C^2| = O(m \log m)$, then $A' = \{A_1'\}$ meets all of the conditions of the theorem. Otherwise we can repeat our argument.

Let R_2 denote the set of rows below the first k_1 rows such that if $r \in R_2$ then r has fewer than $3t-2$ 1's when restricted to C^2, and let C_{R_2} be the set of columns where one of these rows has a 1 in C^2. Let A_2 be A_1 restricted to C^2 after ignoring the rows of R_2 and let k_2 be the largest integer such that $t \cdot I_{k_2} \prec A_2$. Note that we can assume $k_2 \geq 3$.

Lemma 11 $k_2 \leq \frac{1}{2}k_1$.

PROOF: Note that any row r that is part of this $t \cdot I_{k_2}$ must appear above the bottom ℓ rows (as restricted to C^2 the bottom ℓ rows either have fewer than t 1's or they are identically 1). Thus restricted to any C_i^1, r is either identically 0, identically 1 or sparse. We will say that a row r is "mostly 1" restricted to C_i^1 if r is identically 1 or sparse restricted to C_i^1 (i.e. r has fewer than t 0's restricted to these columns). Rearrange rows so that this $t \cdot I_{k_2}$ appears in the first k_2 rows.

Note that because $k_2 \geq 3$, no column can have two 1's in the first k_2 rows. As $|C_i^1| \geq 3t - 2 \geq 2t - 1$ for all i, any two rows that are mostly 1 restricted to any C_i^1 must contain a column with 1's in both of these rows. Hence restricted to any C_i^1 and the first k_2 rows, there can be at most one mostly 1 row.

If row $1 \leq j \leq k_2$ is not mostly 1 when restricted to any C_i^1, then we could use row j to create a $t \cdot I_{k_1+1} \prec A_1$ by swapping it with our original $k_1 + 1$th row, contradicting the definition of k_1. If there is precisely one i such that j restricted to C_i^1 is mostly 1, then swapping row j with the original ith row gives a $t \cdot I_{k_1}$ that would have given us a smaller value for $|C^2|$ (as at least $3t - 2$ 1's get added from C^2 and at most $t - 1$ 1's are replaced by 0's of the mostly 1 row), which contradicts our choice of $t \cdot I_{k_1} \prec A_1$. Hence every row $1 \leq j \leq k_2$ must be mostly 1 restricted to at least two different C_i^1, but as each C_i^1 can only contribute at most one mostly 1 row we must have $k_2 \leq \frac{1}{2}k_1$. □

We then perform identical arguments for the corresponding C_i^2 columns as we did with the C_i^1 columns to get an A_2'. If C^3 is defined analogous to C^2 and if $C^3 = O(m \log m)$, then we can take $A' = \{A_1', A_2'\}$ which satisfies all the conditions of the theorems. If not, we repeat the same argument. But by Lemma 11 this process can continue at most $\log m$ times, and when the process terminates A' excludes only $O(m \log m)$ columns of A_0 (as it ignores $O(m)$ columns at each of the potentially $\log m$ steps), so it meets all of the criteria of the theorem. □

Theorem 6 allows us to reduce computing upper bounds of matrices in $\text{Avoid}(m, \mathcal{F})$ where $Q_3(t) \in \mathcal{F}$ to computing upper bounds of matrices that are of the same form as the A_j' matrices.

Corollary 12 *For $\mathcal{F}$ with $Q_3(t) \in \mathcal{F}$, let $\tilde{A}$ be the largest matrix such that $\tilde{A} \in \text{Avoid}(m, \mathcal{F})$ and such that it meets all the requirements of the A_j' matrices in the statement of Theorem 6. Then* $\text{forb}(m, \mathcal{F}) = O(\max\left\{|\tilde{A}|, m\right\} \log m)$.

PROOF: The statement certainly holds if $\text{forb}(m, \mathcal{F}) = O(m \log m)$. Assume $\text{forb}(m, \mathcal{F}) = \omega(m \log m)$. Then if A is a maximum sized matrix in $\text{Avoid}(m, \mathcal{F})$ we can apply Theorem 6 to get a set of configurations $A' = \left\{A_j'\right\}$ with $|A_j'| \leq |\tilde{A}|$ for all j (as necessarily $A_j' \in \text{Avoid}(m, \mathcal{F})$ since $A_j' \prec A \in \text{Avoid}(m, \mathcal{F})$), and we have $|A| = O(\sum |A_j'|)$ or $|A| = O(|\tilde{A}| \log m)$. □

We suspect that the statement of Corollary 12 can be strengthened to $O(\max\left\{|\tilde{A}|, m\right\})$, but as stated the Corollary can still be used to prove near optimal results. It is possible to get tighter upper bounds for certain configurations by using some of the additional structure provided by Theorem 6.

Theorem 13 *If $s \leq r$ then* $\text{forb}(m, Q_3(t), I_r \times I_s^c) = O(m^{2-1/s})$.

PROOF: We first prove this for the case $t = 2$. Let $A \in \text{Avoid}(m, Q_3(2), I_r \times I_s^c)$ with $|A| = \omega(m \log m)$ and let A' be the corresponding set obtained from Theorem 6. We focus our attention on bounding $|A_1'|$. Note that restricted to C_i^1, there must exist $|C_i^1|$ rows that are distinct rows of $I_{|C_i^1|}^c$ (one to identify each column of C_i^1). Denote a set of such rows by R_i. If there exists a set of integers $\{i_1, \ldots, i_r\}$ such that $|R_{i_1} \cap \cdots \cap R_{i_r}| \geq s$, then by taking these s rows, the rows $i_1, \ldots, i_r$ and the relevant columns we can find an $I_r \times I_s^c$ in A_1' (since we have an I_s^c occurring simultaneously under r different I_{k_1} columns). How large can $|A_1'| = \sum |C_i^1|$ be given this restriction?

We rephrase this problem in terms of graph theory. We form a bipartite graph $G(C, R)$ where $v_i \in C$ for $1 \leq i \leq k_1$ corresponding to the C_i^1 columns, and $r \in R$ corresponding to each row below the

first k_1 rows. G will contain the edge $v_i r$ iff $r \in R_i$. Our restriction of no set $\{i_1, \ldots, i_r\}$ such that $|R_{i_1} \cap \cdots \cap R_{i_r}| \geq s$ means that G does not contain a $K_{r,s}$, the complete bipartite graph with vertex sets of size r and s, with the r vertices coming from C and the s vertices coming from R. Using standard arguments from extremal graph theory, this graph can have at most $c|R||C|^{1-1/s} + d|C| \leq cmk_1^{1/s} + dk_1$ edges for some constants c and d. Hence in total we have that

$$\sum |A'_i| \leq \sum (cmk_i^{1-1/s} + dk_i) \leq cmk_1^{1-1/s} \sum \left(\frac{1}{2}\right)^{i(1-1/s)} + dk_1 \sum \left(\frac{1}{2}\right)^i = O(m^{2-1/s}),$$

and thus this is an asymptotic upper bound for $|A| = \Theta(\sum |A'_i|)$.

We wish to generalize this argument for arbitrary t. The key idea is that for each set C_i^j we must find a set of rows R_i^j with $|R_i^j| = \Theta_t(|C_i^j|)$ and such that R_i^j contains an $I^c_{|R_i^j|}$. Once we have this, we can perform the same graph argument on these R_i^j rows as we did for the R_i rows above and get the same asymptotic results. The following lemma accomplishes this goal by taking $B = C_i^j$ after ignoring rows that are identically 0.

□

Lemma 14 *Given an integer t, let B be a matrix consisting of rows with fewer than t 0's such that every column of B has a 0 in some row. Then there exists a set of rows R of B such that:*

1. *R contains an $I^c_{|R|}$.*
2. *$|R| \geq 2^{2-t}|B|$.*

Proof: The $t = 2$ case is obvious (for every column take a row that has a 0 in the column), so inductively assume the statement holds up to $t-1$. We wish to partition the columns of B into two sets, B_1 and B_2. Remove the leftmost column c of B and add it to B_1, and remove all columns c' where there exists a row r such that r has a 0 in both column c and column c' and add these columns to B_2. Repeat this process until every column of B is in one of these sets, and note that $B_i \geq \frac{1}{2}|B|$ for some i. Note that as every column of B was identified, every column of B_1 and B_2 is also identified.

If $B_1 \geq \frac{1}{2}|B|$, then note that no row r has more than one 0 in B_1 (if r had 0's in $c, c' \in B_1$ with c to the left of c', then c' should have been added to B_2), so by the $t = 2$ case we can find a set R with $|R| = |B_1| \geq \frac{1}{2}|B|$ that contains an $I^c_{|R|}$.

If $|B_2| \geq \frac{1}{2}|B|$, then note that B_2's rows all have at most $t-2$ 0's (as every row with a 0 in some c' originally had a 0 in the corresponding c column from B_1), so by the inductive hypothesis we can find a set R with $|R| \geq 2^{2-(t-1)}|B_2| \geq 2^{2-t}|B|$ that contains an $I^c_{|R|}$. □

We can use the graph idea from the proof of Theorem 13 to achieve lower bounds as well.

Theorem 15 $\text{forb}(m, Q_3(t), I_r \times I_s^c) = \Omega(ex(m, K_{r,s}))$.

Proof: We define a generalized product operation for matrices. Let A and B be simple matrices with m_1 and m_2 rows respectively and $G = G(C_A, C_B)$ a bipartite graph with the vertex set C_A corresponding to the set of columns of A and C_B to the set of columns of B. We define $A \times_G B$ to be the simple matrix on $m_1 + m_2$ rows such that it contains the column defined by placing the column $a \in C_A$ on the column $b \in C_B$ iff $ab \in E(G)$. Thus $|A \times_G B| = |E(G)|$.

Let $G(V, W)$ be a bipartite graph on m vertices such that G avoids $K_{r,s}$ and such that G has the maximum number of edges. Note that using the probabilistic method it is easy to show that $|E(G)| \geq \frac{1}{2}ex(m, K_{r,s})$. We claim that $A = I_{|V|} \times_G I^c_{|W|} \in \text{Avoid}(m, Q_3(t), I_r \times I_s^c)$, and hence $\text{forb}(m, Q_3(t), I_r \times I_s^c) \geq \frac{1}{2}ex(m, K_{r,s})$. We certainly have $Q_3(t) \not\prec A$ as A is a sub-matrix of $I_a \times I_a^c$ for $a = \max\{|V|, |W|\}$, which avoids $Q_3(t)$. Note that if $I_r \times I_s^c \prec A$ Then we must have all of the I_r rows coming entirely from either the $I_{|V|}$ rows of A or the $I^c_{|W|}$ rows and the I_s^c rows coming entirely from the other. Indeed, no

two rows of the $I_{|V|}$ block of A contains a column of two 1's, but every row of I_r in $I_r \times I_s^c$ together with a row of I_s^c contains a column of two 1's, so the $I_{|V|}$ rows can contribute to at most one of these blocks. Further note that if $s \geq 3$ then the I_s^c must come from the $I_{|W|}^c$ block (as it needs a column with two 1's), and similarly if $r \geq 3$ then I_r must come from the $I_{|V|}$ block (and hence again the I_s^c must come from the $I_{|W|}^c$ block).

Now consider $B = I_{|V|} \times_G I_{|W|}$. If $I_r \times I_s^c \prec A$ then we certainly have $I_r \times I_s \prec B$ (if s or r were at least 3 then the I_s^c must have been in $I_{|W|}^c$ and then complimented to become an I_s, and if $s = r = 2$ complimenting either block would still leave you with an $I_2 \times I_2$). But $I_{|V|} \times_G I_{|W|}$ is the incidence matrix of G, a graph that avoids $K_{r,s}$, and hence it must avoid $I_r \times I_s$, the incidence matrix of $K_{r,s}$. Thus we could not have had $I_r \times I_s^c \prec A$.

□

It is known that $ex(m, K_{r,s}) = \Theta(m^{2-1/s})$ for $(s-1)! \leq r$, so for these values of s and r our bounds from Theorems 13 and 15 are sharp. In particular, because $F_{11} = I_2 \times I_2 = I_2 \times I_2^c$, we have the following result.

Corollary 16 $\text{forb}(m, Q_3, F_{11}) = \Theta(m^{3/2})$.

3 Avoiding $1_{k,\ell}$

In this section we study the identically 1 matrices $1_{k,\ell}$. We first note an immediate consequence of Theorem 2.

Corollary 17 $\text{forb}(m, 1_{k,\ell}, F) = \Theta(1)$ *for* $F = I_3,\ F_{10},$ *or* $0_{k,\ell}$.

PROOF: Note that $1_{k,\ell} \prec T_{k+\ell}, I_{k+\ell}^c$ and that $I_3, F_{10} \prec I_4$ and $0_{k,\ell} \prec I_{k+\ell}$. We thus have an upper bound of $BB(k+\ell)$ by Theorem 2. □

We next consider a slight generalization of a result from [7].

Theorem 18 *Let G be the incidence matrix of a $(k-1)$-uniform hypergraph $\mathcal{H}$. Then*

$$\text{forb}(m, 1_{k,1}, G) = \binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{k-2} + ex^{(k-1)}(m, \mathcal{H})$$

PROOF: As a lower bound one can take all columns with fewer than $k-1$ 1's, along with the incidence matrix of a maximum $(k-1)$-uniform $\mathcal{H}$ avoiding hypergraph. For an upper bound, note that one can have at most $\binom{m}{0} + \cdots + \binom{m}{k-2}$ columns with fewer than $k-1$ 1's, and the columns with weight $k-1$ define the incidence matrix of a $(k-1)$-uniform hypergraph that avoids $\mathcal{H}$, and hence can be no larger than $ex^{(k-1)}(m, H)$. □

Corollary 19

$$\text{forb}(m, 1_{k,1}, I_{s_1} \times \cdots I_{s_{k-1}}) = \binom{m}{0} + \cdots + \binom{m}{k-2} + ex(m, K^{(k-1)}(s_1, \ldots, s_{k-1})).$$

In particular, $\text{forb}(m, 1_{3,1}, F_{11}) = 1 + m + ex(m, K_{2,2}) = \Theta(m^{3/2})$.

We can get similar results when considering configurations of the form $1_{k,2}$.

Theorem 20 *Let G be the incidence matrix of a k-uniform complete r-partite hypergraph $\mathcal{H}$ with $r \geq k$. Then*

$$\text{forb}(m, 1_{k,2}, G) = \binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{k-1} + ex^{(k)}(m, \mathcal{H})$$

PROOF: For a lower bound, again take all columns with fewer than k 1's along with the incidence matrix of a maximum $\mathcal{H}$ avoiding k-uniform hypergraph. Let A be a maximum matrix of Avoid$(m, 1_{k,2}, G)$ and let A' be a matrix obtained from A by taking every column with more than k 1's and removing 1's until these columns have k 1's. We claim that $A' \in$ Avoid$(m, 1_{k,2}, G)$. Clearly $1_{k,2} \not\prec A'$ (if $1_{k,2} \not\prec A$ then removing 1's from A can't induce this configuration) and A' is simple (the columns with fewer than k 1's were already distinct, and if any columns with k 1's were identical we would have a $1_{k,2}$), so all that remains is to show that $G \not\prec A'$.

To see this, we claim that if G' is the matrix obtained by changing any 0 of G to a 1 then G' contains a $1_{k,2}$. This claim is equivalent to saying that if one extends any $e \in E(\mathcal{H})$ to $e' = e \cup \{v\}$ for some $v \in V(\mathcal{H}),\ v \notin e$, then there exists an $f \in E(\mathcal{H})$ such that $|e' \cap f| = k$. If e contains no vertices that are in the same partition class as v, then if f is any k-subset of e' that includes v then $f \in E(\mathcal{H})$ and $|e' \cap f| = k$. If e contains a vertex v' that belongs to the same partition class as v, then $f = e' \setminus \{v'\} \in E(\mathcal{H})$ with $|e' \cap f| = k$, and thus we've proven the claim. This means that A can not contain any configuration that is obtained by taking 0's of G and changing them to 1's (since A avoids $1_{k,2}$), and hence the procedure of deleting 1's from A can not induce a G if $G \not\prec A$, so we have $G \not\prec A'$.

Thus for an upper bound of forb$(m, 1_{k,2}, G)$, one only needs to consider matrices where each column has at most k 1's, and this clearly gives the above upper bound. □

Corollary 21

$$\text{forb}(m, 1_{k,2}, I_{s_1} \times \cdots I_{s_k}) = \binom{m}{0} + \cdots + \binom{m}{k-1} + ex(m, K^{(k)}(s_1, \ldots, s_k)).$$

We note that the statement of Theorem 20 is not as strong as possible when $k > 2$. For example, the theorem statement and general proof also applies to the configuration F stated below, despite it not being the incidence matrix of a complete r-partite 3-uniform hypergraph. It would be interesting to know of a complete characterization of k-uniform hypergraphs that satisfy Theorem 20.

$$F = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

Unfortunately for $\ell > 2$, this "downgrading" technique no longer works. We are, however, able to obtain some partial results.

Theorem 22 *For $\ell > 2$,*

$$\text{forb}(m, 1_{k,\ell}, I_{s_1} \times \cdots \times I_{s_k}) = \Omega(ex^{(k)}(m, K(s_1, \ldots, s_k)))$$

$$\text{forb}(m, 1_{k,\ell}, I_{s_1} \times \cdots \times I_{s_k}) = O(ex^{(k)}(m, K(s_1 + c_1, \ldots, s_k + c_k))),$$

where $c_i = (\ell - 1) \max_{j \neq i} \left\{ \frac{s_j - 1}{2} \right\} \prod_{j \neq i} s_j$.

We believe that this can be improved to forb$(m, 1_{k,\ell}, I_{s_1} \times \cdots \times I_{s_k}) = \Theta(ex^{(k)}(m, K(s_1, \ldots, s_k)))$, though we are unable to do so here. Nevertheless, $ex^{(k)}(m, K(s_1 + c_1, \ldots, s_k + c_k)) = o(m^k)$, so this bound is non-trivial.

PROOF: The lower bound is simply the incidence matrix of the extremal hypergraph. We first prove the upper bound for $k = 2$ to demonstrate the general idea of the proof. Let A be a maximum matrix in Avoid$(m, 1_{2,\ell}, I_r \times I_s)$ that has no columns with fewer than two 1's (and hence the forb function will be at most $O(m)$ larger than $|A|$). Let C_i denote the set of columns of A whose first 1 is in row i. Note that any row $j \neq i$ restricted to C_i has at most $\ell - 1$ 1's (otherwise the row together with the ith would induce a $1_{2,\ell}$), and further note that each column of C_i has a 1 in some row other than the ith (since every

column has at least two 1's), i.e. every column of C_i is identified by a 1. We can thus use Lemma 14 (after switching 0's and 1's in the lemma statement) to find a set of rows R_i such that restricted to C_i these rows contain a $I_{|R_i|}$ and such that $|R_i| \geq 2^{2-\ell}|C_i|$. We then define a bipartite graph with one vertex set corresponding to the C_i column sets and the other vertex set corresponding to the rows of A, and we draw an edge between C_i and r if $r \in R_i$. We would like to say that if this graph contains a $K_{r,s}$ (say the r vertices coming from the C_i vertex set and the s vertices coming from the R_i vertex set, which is a non-trivial assumption we will deal with later), then A contains an $I_r \times I_s$, but this isn't quite the case. For example, if

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix},$$

then A does not contain a $I_2 \times I_2$, despite the corresponding graph being $K_{2,2}$. The problem is that if we want to use columns from C_i and $C_{i'}$ with $i < i'$, it's possible that there are 1's in the i'th row of C_i, and if these 1 columns correspond with the I_s under C_i then we can't actually use these columns. Fortunately, each row below the ith row of C_i contains fewer than ℓ 1's, so this problem can't happen too many times. We claim that if instead of having an I_s simultaneously under r different C_i we had an I_{s+c_2}, where $c_2 = (\ell-1)\frac{r(r-1)}{2}$, simultaneously under r different C_i, then we could find an $I_r \times I_s$.

Assume that we have this situation with the i's of our C_i's belonging to the set $\{i_1, \ldots, i_r\}_<$, and let R'_0 denotes the set of rows that contain the simultaneous I_{s+c_2} under these C_i, noting that $|R'_0| = s + (\ell-1)\frac{r(r-1)}{2}$. For $r \in R'_0$, we will say that its corresponding column restricted to C_{i_j} is the column where r contains the 1 it contributes to the $I_{|R'_0|}$ in C_{i_j}. Note that restricted to the $r-1$ rows $\{i_2, \ldots, i_r\}$, C_{i_1} contains at most $(\ell-1)(r-1)$ 1's (as each row has at most $\ell-1$ 1's). Thus if B_1 is the set of columns of C_{i_1} with 1's in these rows we have $|B_1| \leq (\ell-1)(r-1)$. Define $R'_1 \subseteq R'_0$ to be the set of rows that have corresponding columns in C_{i_1} that are not in B_1, and hence $|R'_1| \geq |R'_0| - (\ell-1)(r-1) = s + (\ell-1)\frac{(r-1)(r-2)}{2}$. Note that restricted to the corresponding columns of R'_1 and the rows $\{i_2, \ldots, i_r\}$, C_{i_1} is identically 0. We can similarly define the subset $R'_2 \subseteq R'_1$ consisting of the rows whose corresponding columns in C_{i_2} are 0 in the rows $\{i_3, \ldots, i_r\}$ (row i_1 is automatically identically 0 restricted to C_{i_2} since $i_1 < i_2$) with $|R'_2| \geq |R'_1| - (\ell-1)(r-2) \geq s + (\ell-1)\frac{(r-2)(r-3)}{2}$. We repeat this process until we reach the set R'_r which satisfies $|R'_r| \geq s$ and under each C_{i_j}, the corresponding columns of R'_r are identically 0 in the other $i_{j'}$ rows. This gives an $I_r \times I_s$.

However, to guarantee an $I_r \times I_s$ in A it is insufficient to simply guarantee the existence of a $K_{r,s+c_2}$ in the graph we constructed, since we could have the $s+c_2$ vertices coming from the C_i vertex set instead of the row vertex set. To remedy this, we must increase r by a suitable amount as well, namely by $c_1 = (\ell-1)\frac{s(s-1)}{2}$, as in this case a symmetric argument will guarantee our result. Thus the existence of a $K_{r+c_1,s+c_2}$ in this graph guarantees an $I_r \times I_s$, so the graph must have $O(ex(m, K_{r+c_1,s+c_2}))$ edges, and hence $|A| = O(ex(m, K_{r+c_1,s+c_2}))$ as well.

For the general problem, again consider a maximum A with every column having at least k 1's and define the set $C(i_1, \ldots, i_{k-1})$ to be the columns which have their first $k-1$ 1's in rows $i_1, \ldots, i_{k-1}$ and with $i_j > i_{j-1}$. Again we can find rows $R(i_1, \ldots, i_{k-1})$ such that the number of rows is proportional to the number of columns of $C(i_1, \ldots, i_{k-1})$, and restricted to these rows and columns there is a large identity matrix. We can then define a k-uniform k-partite hypergraph with vertex sets V_j for $1 \leq j < k$ corresponding to all possible choices of i_j, and vertex set V_k corresponding to all rows of A. We then add the hyperedge $\{i_1, \ldots, i_{k-1}, r\}$ to our hypergraph iff $r \in R(i_1, \ldots, i_{k-1})$. If this hypergraph contains a $K^{(k)}(s_1 + c_1, \ldots, s_k + c_k)$ where $c_i = (\ell-1)\max_{j \neq i}\left\{\frac{s_j-1}{2}\right\}\prod_{j \neq i} s_j$, then we claim that A contains a $I_{s_1} \times \cdots \times I_{s_k}$.

Assume that this hypergraph contains a $K^{(k)}(s_1 + c_1, \ldots, s_k + c_k)$, say on the vertex sets $V'_1, \ldots, V'_k$ with $V'_j \subseteq V_j$ and $|V'_i| = s_i + c_i$ (again, an assumption we'll have to address later). First note that if $i_j \in V'_j$ and $i_{j'} \in V'_{j'}$ with $j < j'$, then $i_j < i_{j'}$. Indeed, because we have a complete k-partite hypergraph, $i_j \in V'_j$ and $i_{j'} \in V'_{j'}$ means that there exists an edge containing both i_j and $i_{j'}$ from these vertex sets. If $j' < k$ then this edge corresponds to a column whose jth 1 is in row i_j and j'th 1 is in row $i_{j'}$,and

if $j < j'$ this only makes sense if $i_j < i_{j'}$. If $j' = k$ then the $i_{j'}$th row must come after the rows where this column has its first $k-1$ 1's by definition, and hence again $i_j < i_{j'}$. This means that for any $C(i_1, \ldots, i_{k-1})$, $i \in V_j$ with $i \neq i_j$ and $j < k-1$, the ith row of $C(i_1, \ldots, i_{k-1})$ is identically 0 (since its $(j+1)$th row with a 1 in it comes from row $i_{j+1} > i$ and its $(j-1)$th comes from $i_{j-1} < i$ if $j \neq 1$), and hence when choosing corresponding rows from V'_k the only potential pitfall will be the rows from V'_{k-1} (as it is possible for $C(i_1, \ldots, i_{k-1})$ to have 1's in row $i \neq i_{k-1}$ even if $i \in V'_{k-1}$).

For $j < k$ let $V''_j \subseteq V'_j$ be any subset with $|V''_j| = s_j$ and let R'_0 be the set of rows corresponding to the $I_{s_k+c_k}$ simultaneously under all of the $C(i_1, \ldots, i_{k-1})$ columns with $i_j \in V''_j$, and we emphasize that our observations in the preceding paragraph shows us that the rows of R'_0 lie entirely below the rows of every V''_j for $1 \leq j < k-1$. Let $i_1, \ldots, i_{k-2}$ be any fixed elements from the V''_j's. Restricted to the columns $C(i_1, \ldots, i_{k-1})$, where i_{k-1} varies amongst all V''_{k-1}, we perform the same procedure that we used for the $k = 2$ case to obtain a set of rows R'_1, after removing at most $(\ell-1)\frac{s_{k-1}(s_{k-1}-1)}{2}$ rows from R'_0, such that that for any $i_{k-1} \in V''_{k-1}$ and any corresponding column of R'_1 restricted to the rows $V''_{k-1} \setminus \{i_{k-1}\}$, $C(i_1, \ldots, i_{k-1})$ is identically 0. We then repeat this process for all possible sequences of $i_1, \ldots, i_{k-2}$, in total removing at most $\frac{s_{k-1}(s_{k-1}-1)}{2} \prod_{j<k-1} s_j$ rows (which in the worst case scenario is $(\ell-1)\max_{j \neq k}\left\{\frac{s_j-1}{2}\right\}\prod_{j \neq k} s_j$). In the end we are left with a set $R' \subseteq R'_0$ with $|R'| \geq s_k$ and in the corresponding columns of any $C(i_1, \ldots, i_{k-1})$ for $i_j \in V''_j$ and restricted to the rows $V''_{k-1} \setminus \{i_{k-1}\}$ the matrix is identically 0. This gives an $I_{s_1} \times \cdots \times I_{s_k}$ in A. Hence the hypergraph can have at most $ex^{(k)}(m, K^{(k)}(s_1+c_1, \ldots, s_k+c_k))$ edges, which means that overall $|A| = O(ex^{(k)}(m, K^{(k)}(s_1+c_1, \ldots, s_k+c_k)))$. $\square$ Next we consider $\mathrm{forb}(m, 1_{4,1}, F_{11})$.

Proposition 23 $\mathrm{forb}(m, 1_{4,1}, F_{11}) = \Theta(m^{3/2})$.

Proposition 23 is a corollary of the following theorem.

Theorem 24 *$r \geq s \geq k-2 \geq 1$ be fixed integers. Then* $\mathrm{forb}(m, 1_{k,1}, I_r \times I_s) = O(m^{k-1-\frac{1}{s}\binom{k-1}{2}})$. *Furthermore, if $r \geq (s-1)!+1$ and $s \geq 2k-4$, then* $\mathrm{forb}(m, 1_{k,1}, I_r \times I_s) = \Theta(m^{k-1-\frac{1}{s}\binom{k-1}{2}})$

For the proof we will apply the kernel (Δ-system) method of Füredi [14]. Let $\mathcal{F}$ be a k-uniform set system on $\{1, 2, \ldots m\}$, furthermore let $\mathcal{M}(F, \mathcal{F}) = \{F \cap F' \colon F \neq F' \in \mathcal{F}\}$. For a k-partite k-uniform system with partite classes $V_1, V_2, \ldots, V_k$ a projection $\pi \colon V_1 \cup V_2 \cup \ldots \cup V_k \to \{1, 2, \ldots, \}$ is defined by $\pi(x) = i \iff x \in V_i$. For a subset $A \subseteq V_1 \cup V_2 \cup \ldots \cup V_k$ define $\pi(A) = \{\pi(a) \colon a \in A\}$. Let us recall that a t-star with kernel X is a collection of t sets $F_1, F_2, \ldots F_t$, such that $F_i \cap F_j = X$ for all $1 \leq i < j \leq t$. Füredi proved the following.

Theorem 25 *For any positive integers $k < t$, there exists a positive real number $c = c(k, t)$ with the following property: If $\mathcal{F}$ is a k-uniform hypergraph, then we can choose a subsystem $\mathcal{F}^* \subset \mathcal{F}$ such that:*

1. *$|\mathcal{F}^*| > c|\mathcal{F}|$.*
2. *Every pairwise intersection in $\mathcal{F}^*$ is a kernel of a t-star of $\mathcal{F}^*$.*
3. *$\mathcal{F}^*$ is k-partite with partite classes $V_1, V_2, \ldots, V_k$.*
4. *There exists a set system $\mathcal{M}$ on $\{1, 2, \ldots, k\}$ such that $\pi(\mathcal{M}(F, \mathcal{F}^*)) = \mathcal{M}$ for all $F \in \mathcal{F}^*$ Furthermore, $\mathcal{M}$ is closed under intersection.*

We also need the following theorem of Alon and Shikhelman. Let $ex(m, G, H)$ mean the largest possible number of subgraphs isomorphic to G in an m-vertex graph that does not have H as subgraph. Alon and Shikhelman prove

Theorem 26 (Alon and Shikhelman) *Let $r \geq s \geq k-1$ be fixed integers. Then $ex(m, K_k, K_{r,s}) = O(m^{k-\frac{1}{s}\binom{k}{2}})$, furthermore, if $r \geq (s-1)!+1$ and $s \geq 2k-2$, then $ex(m, K_k, K_{r,s}) = \Theta(m^{k-\frac{1}{s}\binom{k}{2}})$.*

PROOF:[Proof of Theorem 24] Work by induction on k, with the base case $k = 3$. Any $A \in \text{Avoid}(m, 1_{3,1}, I_r \times I_s)$ has columns of sum at most 2. The columns of sum exactly two form the vertex-edge incidence matrix of a graph that does not contain $K_{r,s}$ as a subgraph, so $|A| \leq 1 + m + ex(m, K_{r,s}) = O(m^{2-1/s})$. Now let $A \in \text{Avoid}(m, 1_{k+1,1}, I_r \times I_s)$ and let A' consist of columns of A of weight k. By the induction hypothesis, $|A \setminus A'| \leq \text{forb}(m, 1_{k,1}, I_r \times I_s) = O(m^{k-1-\frac{1}{s}\binom{k-1}{2}})$. Consider columns of A' as characteristic vectors of a k-uniform hypergraph $\mathcal{F}$. Let $t = r + s$ and let $\mathcal{F}^*$ be the k-partite subhypergraph of $\mathcal{F}$ given by Theorem 25, with partite classes $V_1, \ldots, V_k$, Let $\mathcal{H}_i$ be the $(k-1)$-partite hypergraph induced by $\mathcal{F}^*$ between the V_j for all $j \neq i$. Observe that $\mathcal{H}_i$ does not contain $I_r \times I_s$ as a trace and hence $|E(\mathcal{H}_i)| \leq \text{forb}(m, 1_{k,1}, I_r \times I_s)$. Consider the set system $\mathcal{M}$ on $\{1, 2, \ldots, k\}$ given by Theorem 25. If there is a $1 \leq i \leq k$ such that $\{1, 2, \ldots, k\} \setminus \{i\} \notin \mathcal{M}$, then $F \cap \bigcup_{j \neq i} V_j$ is not an intersection in $\mathcal{F}^*$ for any $F \in \mathcal{F}^*$. This implies that $|\mathcal{F}^*| \leq |E(\mathcal{H}_i)| \leq \text{forb}(m, 1_{k,1}, I_r \times I_s)$. Otherwise, every $k-1$ subset of $\{1, 2, \ldots, k\}$ is in $\mathcal{M}$, and as $\mathcal{M}$ is closed under intersections, $\mathcal{M}$ contains every pair. We claim that the 2-shadow of $\mathcal{F}^*$ does not contain $K_{r,s}$ as a subgraph. Indeed, suppose thare is a $K_{r,s}$ in the 2-shadow and let $\{x_1, x_2\}$ be one of its edges. Thus there is an $F_1 \in \mathcal{F}^*$ such that $\{x_1, x_2\} \subset F_1$. Since $\mathcal{M}$ contains every pair, $\{x_1, x_2\} \in \mathcal{M}(F_1, \mathcal{F}^*)$ and thus it must be the kernel of a t-star $\{F_1, F_2, \ldots, F_t\}$ in $\mathcal{F}^*$. However, this implies that at least one of the F_i's is disjoint from $V(K_{r,s}) \setminus \{x_1, x_2\}$. Applying this to every edge of $K_{r,s}$ we obtain that $\mathcal{F}^*$ has $K_{r,s}$ as a *trace*, that is A' has $I_r \times I_s$ as a configuration. Thus, we inferred that the 2-shadow does not have $K_{r,s}$ as a subgraph. Apply Theorem 26 to the graph determined by the 2-shadow of $\mathcal{F}^*$ and obtain that the number of K_k subgraphs is at most $O(m^{k-\frac{1}{s}\binom{k}{2}})$, which clearly is an upper bound for $|\mathcal{F}^*|$.

To prove the lower bound take a graph G that gives the lower bound in Alon-Shikhelman' Theorem and let $\mathcal{F}$ consists of those k-subsets of the vertices that induce a complete graph. Since G does not have $K_{r,s}$ subgraph, $\mathcal{F}$ does not have $K_{r,s}$ as *trace*, so if A is the vertex-edge incidence matrix of $\mathcal{F}$, then $A \in \text{Avoid}(m, 1_{k+1,1}, I_r \times I_s)$. □ Note that the upper bound in Proposition 23 is obtained by putting $r = s = k - 1 = 2$. The lower bound in Theorem 24 does not give the lower bound of Proposition 23 directly, however the vertex-edge incidence matrix of a maximal C_4-free grah works.

Remark 27 *Despite the largest product avoiding* 1_4 *and* $I_r \times I_s$ *being a 1-fold product, Theorem 24 shows that one can make* $\text{forb}(m, 1_4, I_r \times I_s) = \Theta(m^{3-\epsilon})$. *Thus the best we could hope for as an extension of Conjecture 1 for general forbidden families is* $\text{forb}(m, F, G) = o(m^p)$ *if* $\text{forb}(m, F) = \Theta(m^p)$ *and there exists no p-fold product avoiding both* F *and* G. *However, we do not dare to formulate this as a conjecture.*

References

[1] N. ALON, AND C. SHIKHELMAN, Many T copies in H-free graphs, *Journal of Combinatorial Theory, Series B* **121** (2016) 146–172.

[2] R.P. ANSTEE, Some problems concerning forbidden configurations, *preprint* (1990).

[3] R.P. ANSTEE, A Survey of forbidden configurations results, *Elec. J. of Combinatorics* **20** (2013), DS20, 56pp.

[4] R.P. ANSTEE, F. BAREKAT, AND A. SALI, Small forbidden configurations V: Exact bounds for 4×2 cases, *Studia. Sci. Math. Hun.* **48** (2011), 1-22.

[5] R.P ANSTEE, C. KOCH,M. RAGGI, AND A. SALI, Forbidden configurations and product constructions, *Graphs and Combinatorics*, **30(6)**, (2014) 1325–1349.

[6] R.P. ANSTEE, AND P. KEEVASH, Pairwise intersections and forbidden configurations. *European Journal of Combinatorics*, **27(8)**, 2006, 1235-1248.

[7] R.P. ANSTEE, C.L. KOCH, Forbidden Families of Configurations, *Australasian J. of Combinatorics*, accepted Nov 2013. 18pp arXiv preprint arXiv:1307.1148, 2013.

[8] R.P. Anstee and Linyuan Lu, Multicoloured Families of Configurations, arXiv:1409.4123, 16pp.

[9] R.P. Anstee, M. Raggi and A. Sali, Forbidden configurations: Boundary cases, *European Journal of Combinatorics* **35** 51-66

[10] R.P. Anstee, A. Sali, Small Forbidden Configurations IV, *Combinatorica* **25**(2005), 503–518.

[11] J. Balogh, B. Bollobás, Unavoidable Traces of Set Systems, *Combinatorica*, **25** (2005), 633–643.

[12] P. Erdős and M. Simonovits, A limit theorem in graph theory. *Studia Sci. Math. Hungar* **1** (1966) 51–57.

[13] P. Erdős, A.H. Stone, On the Structure of Linear Graphs, *Bull. A.M.S.*, **52**(1946), 1089–1091.

[14] Z. Füredi, On finite set-systems whose every intersection is a kernel of a star. *Discrete mathematics*, **47**, (1983) 129-132.

Regret Ratio Minimization in Multi-objective Submodular Function Maximization

Tasuku Soma[1]

The University of Tokyo,
Tokyo, Japan.
tasuku_soma@mist.i.u-tokyo.ac.jp

Yuichi Yoshida[2]

National Institute of Informatics, *and*
Preferred Infrastructure, Inc.,
Tokyo, Japan.
yyoshida@nii.ac.jp

Abstract: Submodular function maximization has numerous applications in machine learning and artificial intelligence. Many real applications require multiple submodular objective functions to be maximized, and it is not known in advance which of the objective functions is regarded to be important by a user. In such cases, it is desirable to have a small family of representative solutions that would satisfy any user's preference. A traditional approach for solving such a problem is to enumerate the Pareto optimal solutions. However, owing to the massive number of Pareto optimal solutions (possibly exponentially many), it is difficult for a user to select a solution. In this paper, we propose methods for finding a small family of representative solutions, based on the notion of regret ratio. The first method outputs a family of fixed size with a non-trivial regret ratio. The second method enables us to choose the size of the output family, and in the biobjective case, it has a provable trade-off between the size and the regret ratio. The last method finds a family of polynomial size with almost optimal regret ratio.

Keywords: Submodular Function, Approximation Algorithm, Multi-objective Optimization

Submodular function maximization has numerous applications in machine learning and artificial intelligence, such as budget allocation [15], document summarization [10, 11], maximum entropy sampling [5], online service privacy [7], and sensor placement [9]. Many efficient algorithms have been developed to solve these problems by maximizing a single submodular function.

However, in real applications, we often face multiple conflicting criteria. For example, in data summarization, we are to select a subset of a data set that maximizes two criteria: coverage and diversity. That is, we are to find a subset that explains the entire data well, and at the same time, elements in the subset are different to each other. Further, in the budget allocation problem, we are to buy ads to maximize the expected number of people influenced by ads, while we also need to minimize the cost of buying ads. These problems prompt us to consider maximizing *multiple* submodular functions.

In contrast to maximizing a single submodular function, maximizing multiple submodular functions is not well understood. The difficulty in multi-objective optimization arises from the fact that there may be no single solution that maximizes all the objective functions simultaneously. Hence, preferable solutions can vary from one user to another, depending on which objective function is more important to the user. Moreover, a user often cannot describe his/her own preference explicitly but can only compare two solutions based on his/her implicit preference. In such cases, a natural goal is to precompute a family of "representative" solutions so that a user with any preference can find an (almost) optimal set in the family.

A standard approach for finding such a family is to enumerate the Pareto optimal solutions. However, this approach has two drawbacks: (i) The number of Pareto optimal solutions is often massive, and

[1]T. S. is supported by JSPS Grant-in-Aid for Research Activity Start-up.

[2]Y. Y. is supported by JSPS Grant-in-Aid for Young Scientists (B) (No. 26730009), MEXT Grant-in-Aid for Scientific Research on Innovative Areas (No. 24106003), and JST, ERATO, Kawarabayashi Large Graph Project.

enumerating all of them does not enable a user to select a solution. (ii) No efficient algorithm for computing the Pareto optimal solutions is known when the objective functions are submodular.

1 Our contributions

In this paper, we tackle the above-mentioned problem using the concept of *regret ratio*, introduced in [13]. Here, we assume that the preference of a user can be expressed as a convex combination of the objective functions. Then, intuitively speaking, the regret ratio of a family of solutions is the (normalized) loss caused by choosing a solution from the family instead of considering all feasible solutions. The advantage of introducing such a concept and optimizing it is that we can control the size of the family.

In this paper, we formalize the concept of regret ratio for multi-objective submodular function maximization. Then, to find a family of solutions with a small regret ratio, we propose three methods, namely the coordinate-wise maximum method, the polytope method, and the reduction method. The coordinate-wise maximum method outputs a family of fixed size with a non-trivial regret ratio. The polytope method enables us to choose the size of the output family, and in the biobjective case, it has a provable trade-off between the size and the regret ratio. The reduction method reduces the problem to a simpler problem in which possible combinations of objective values are polynomially many and explicitly given. This method runs in polynomial time and attains almost optimal regret ratio if the number of objectives is a constant. All the methods can handle monotone and non-monotone submodular functions under any constraint as long as there is an approximation algorithm for the corresponding problem on a *single* submodular function.

2 Related Work

The notion of regret ratio was originally introduced for obtaining a subset of representative points from a point set [13]. Several notions of representative sets have been proposed, including *k-representative skyline queries* [12, 16], *top-k dominating queries* [18], and *ϵ-skyline queries* [17]. In comparison to these notions, regret ratio has the following desirable properties: (i) *scale invariance*, i.e., even if we multiply the values of some coordinate by a positive constant, the regret ratio of a set remains unchanged; (ii) *stability*, i.e., adding a point that is unimportant, in the sense that it is not optimal for any preference, does not change the regret ratio of a set; (iii) *parameter-freeness*, i.e., only the number of points to be selected is required. These features strongly motivate us to compute a family of solutions with a small regret ratio in the multi-criteria setting.

We note that for point sets, there are algorithms with a provable trade-off between the size of the output set and its regret ratio [1, 13, 14]. However, these algorithms cannot be directly applied to our submodular setting because they check all the points, which takes exponential time in our setting.

In a similar problem, the robust submodular function maximization problem [6], multiple monotone submodular functions $f_1, \ldots, f_d : 2^E \to \mathbb{R}_+$ and an integer k are given; the goal is to find a set $S \subseteq E$ of size at most k that maximizes $\min\{f_1(S), \ldots, f_d(S)\}$. In our problem, we consider a (unknown) convex combination of $f_1, \ldots, f_d$ and output a family of sets instead of a single set.

The linear submodular bandit problem [19, 8] also considers convex combinations of submodular objective functions. In this problem, convex coefficients are drawn from some unknown distribution and one can learn the distribution with sampling and optimization. On the other hand, our setting is *adversarial* in the sense that we must consider all possible convex combinations.

3 Preliminaries

For an integer k, let $[k]$ denote the set $\{1, 2, \ldots, k\}$. We denote the set of nonnegative reals by $\mathbb{R}_+$. Let E be a finite ground set. A function $f : 2^E \to \mathbb{R}$ is said to be *submodular* if

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$$

for every $X, Y \subseteq E$. It is well known that submodularity is equivalent to the *diminishing return property*: $f(X \cup \{e\}) - f(X) \geq f(Y \cup \{e\}) - f(Y)$ for every $X \subseteq Y \subsetneq E$ and $e \in E \setminus Y$.

For functions $f_1, \ldots, f_d : 2^E \to \mathbb{R}$ and a vector $\boldsymbol{a} \in \mathbb{R}^d$, we define a function $f_{\boldsymbol{a}}(X) := \sum_{i=1}^{d} a(i) f_i(X)$. Note that, if $f_1, \ldots, f_d$ are submodular and $\boldsymbol{a} \in \mathbb{R}^d_+$, then $f_{\boldsymbol{a}}$ is also submodular.

3.1 Regret-minimizing family

Let $\mathcal{C} \subseteq 2^E$ be a family of sets, which we regard as a constraint on solutions. Let $\mathcal{S} \subseteq \mathcal{C}$ be a subfamily of $\mathcal{C}$ and $f : 2^E \to \mathbb{R}_+$ be a function. We define the *regret* of $\mathcal{S}$ with respect to f under the constraint $\mathcal{C}$ as $r_{f,\mathcal{C}}(\mathcal{S}) := \max_{X \in \mathcal{C}} f(X) - \max_{X \in \mathcal{S}} f(X)$. Then, we define the regret ratio of $\mathcal{S}$ with respect to f under $\mathcal{C}$ as

$$\mathrm{rr}_{f,\mathcal{C}}(\mathcal{S}) = \frac{r_{f,\mathcal{C}}(\mathcal{S})}{\max_{X \in \mathcal{C}} f(X)} = 1 - \frac{\max_{X \in \mathcal{S}} f(X)}{\max_{X \in \mathcal{C}} f(X)}.$$

Note that $\mathrm{rr}_{f,\mathcal{C}} \in [0, 1]$ and that $\mathrm{rr}_{f,\mathcal{C}}(\mathcal{S})$ represents the normalized loss caused by choosing a solution from $\mathcal{S}$ instead of $\mathcal{C}$. Then, the *(maximum) regret ratio* of $\mathcal{S}$ with respect to functions $f_1, \ldots, f_d : 2^E \to \mathbb{R}_+$ is defined as

$$\mathrm{rr}_{f_1,\ldots,f_d,\mathcal{C}}(\mathcal{S}) = \max_{\boldsymbol{a} \in \mathbb{R}^d_+} \mathrm{rr}_{f_{\boldsymbol{a}},\mathcal{C}}(\mathcal{S}).$$

Intuitively speaking, $\boldsymbol{a} \in \mathbb{R}^d_+$ represents a preference of a user on the functions $f_1, \ldots, f_d$, and $\mathrm{rr}_{f_1,\ldots,f_d,\mathcal{C}}(\mathcal{S})$ is the worst regret ratio over all the preferences. We often omit the subscripts of $f_1, \ldots, f_d$ when they are clear from the context.

We study the following problem in this paper:

Definition 1 (Regret ratio minimization in multi-objective submodular function maximization) *Given submodular functions $f_1, \ldots, f_d : 2^E \to \mathbb{R}_+$, $\mathcal{C} \subseteq 2^E$, and $k \geq d$, find $\mathcal{S} \subseteq \mathcal{C}$ with $|\mathcal{S}| \leq k$ that minimizes the maximum regret ratio $\mathrm{rr}_{f_1,\ldots,f_d,\mathcal{C}}(\mathcal{S})$.*

If $\boldsymbol{a}$ is fixed, finding $X^* \in \mathcal{C}$ that maximizes $f_{\boldsymbol{a}}(X)$ is called *submodular function maximization*, which is an NP-hard problem in general. However, for various constraint families $\mathcal{C}$, one can find an approximate solution efficiently. If one can find an α-approximate solution X^*, the corresponding regret ratio is $1 - \frac{f_{\boldsymbol{a}}(X^*)}{\max_{X \in \mathcal{C}} f(X)} \leq 1 - \alpha$.

3.2 Geometric Interpretation

The multi-objective submodular function maximization has a nice geometric interpretation. Let us consider a function $\boldsymbol{f}(X) := [f_1(X) \ldots f_d(X)]^\top \in \mathbb{R}^d_+$. Note that $f_{\boldsymbol{a}}(X) = \boldsymbol{a}^\top \boldsymbol{f}(X)$. For $\mathcal{S} \subseteq \mathcal{C}$, we define $C_{\boldsymbol{f}}(\mathcal{S}) := \mathrm{conv}\{\boldsymbol{f}(X) : X \in \mathcal{S}\}$. We associate $\mathcal{S} \subseteq \mathcal{C}$ with a polytope

$$P(\mathcal{S}) := \{\boldsymbol{x} \in \mathbb{R}^d_+ : \text{there exists } \boldsymbol{y} \in C_{\boldsymbol{f}}(\mathcal{S}) \text{ such that } \boldsymbol{x} \leq \boldsymbol{y}\},$$

where $\boldsymbol{x} \leq \boldsymbol{y}$ means $x(i) \leq y(i)$ $(i \in [d])$.

Lemma 2 ([14, Lemma 1]) *$\mathrm{rr}_{f_1,\ldots,f_d,\mathcal{C}}(\mathcal{S}) \leq 1 - \alpha$ if and only if $P(\mathcal{C}) \subseteq \alpha^{-1} P(\mathcal{S})$.*

The above characterization establishes that the maximum regret ratio is *scale-invariant*, i.e., even if we replace f_i with βf_i for some $\beta > 0$, the regret ratio is preserved. The following lemma is just a restatement of the above lemma, but is useful for the analysis of our algorithms. A *frontier* face is a face of $P(\mathcal{S})$ consisting of Pareto optimal points.

Lemma 3 *$\mathrm{rr}_{f_1,\ldots,f_d,\mathcal{C}}(\mathcal{S}) = \max_{\boldsymbol{a}} \mathrm{rr}_{f_{\boldsymbol{a}},\mathcal{C}}(\mathcal{S})$, where $\boldsymbol{a}$ runs over the nonnegative normal vectors of all frontier faces of $P(\mathcal{S})$.*

4 Algorithms

In this section, we present three algorithms. These algorithms require approximation algorithms for maximizing submodular functions. Let α be the minimum approximation ratio of these approximation algorithms. The first algorithm, the coordinate-wise maximum method, always outputs a family of d solutions with regret ratio $1-\alpha/d$. The second algorithm, the polytope method, has a provable guarantee only when $d = 2$. However, it has a trade-off between the regret ratio and the size of the output, and the regret ratio converges to $1-\alpha$ as the output size increases. The last algorithm, the reduction method, has almost optimal regret ratio dependence on k and runs in polynomial time. Note that our algorithms do not quite depend on submodularity, but exploit the fact that the nonnegative combination of submodular functions is again submodular.

4.1 Coordinate-wise maximum method

Besides functions $f_1, \ldots, f_d : 2^E \to \mathbb{R}_+$ and a constraint $\mathcal{C} \subseteq 2^E$, the coordinate-wise maximum method requires an approximation algorithm $\mathcal{A}_i$ for $\max_{X\in\mathcal{C}} f_i(X)$ $(i \in [d])$. Then, it simply computes an approximate solution X_i for $\max_{X\in\mathcal{C}} f_i(X)$ by using $\mathcal{A}_i$ for each $i \in [d]$, and subsequently outputs $\mathcal{S}_{\mathrm{coord}} := \{X_1, \ldots, X_d\}$. See Algorithm 1 for further details.

Algorithm 1 Coordinate-wise maximum method

Require: Submodular functions $f_1, \ldots, f_d : 2^E \to \mathbb{R}_+$, a constraint $\mathcal{C} \subseteq 2^E$, and an approximation algorithm $\mathcal{A}_i$ for $\max_{X\in\mathcal{C}} f_i(X)$ $(i \in [d])$.

1: **for** $i \in [d]$ **do**
2: $\quad X_i \leftarrow$ a solution obtained by applying $\mathcal{A}_i$ to f_i.
3: **return** $\mathcal{S}_{\mathrm{coord}} := \{X_1, \ldots, X_d\}$.

Lemma 4 *Let α be the minimum approximation ratio of $\mathcal{A}_i$'s. Then, we have $\mathrm{rr}_{\mathcal{C}}(\mathcal{S}_{\mathrm{coord}}) \leq 1 - \frac{\alpha}{d}$.*

Proof: For any $\boldsymbol{a} \in \mathbb{R}^d_+$, we have

$$\begin{aligned}
\max_{i\in[d]} f_{\boldsymbol{a}}(X_i) &\geq \frac{1}{d}\sum_{i\in[d]} a(i) f_i(X_i) \\
&\geq \frac{\alpha}{d}\sum_{i\in[d]} a(i) \max_{X\in\mathcal{C}} f_i(X) \\
&\geq \frac{\alpha}{d}\max_{X\in\mathcal{C}} \sum_{i\in[d]} a(i) f_i(X) \\
&= \frac{\alpha}{d}\max_{X\in\mathcal{C}} f_{\boldsymbol{a}}(X).
\end{aligned}$$

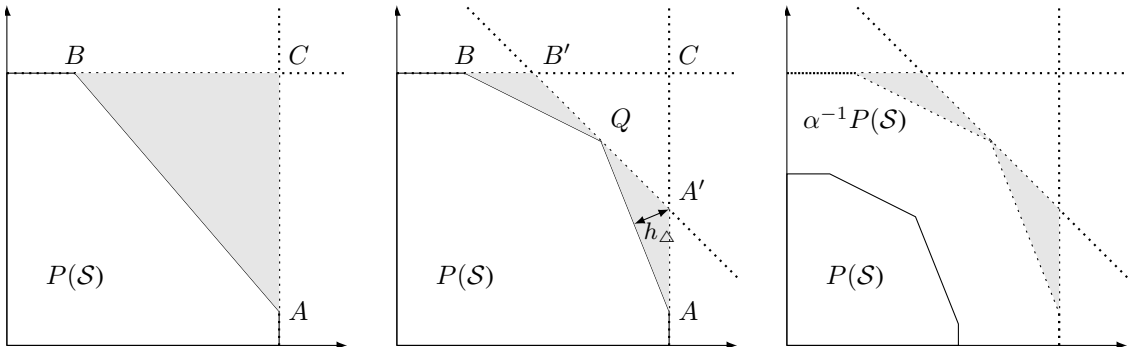
Therefore, we have

$$\mathrm{rr}_{\mathcal{C}}(\mathcal{S}_{\mathrm{coord}}) = \max_{\boldsymbol{a}\in\mathbb{R}^d_+} \left[1 - \frac{\max_{i\in[d]} f_{\boldsymbol{a}}(X_i)}{\max_{X\in\mathcal{C}} f_{\boldsymbol{a}}(X)}\right] \leq 1 - \frac{\alpha}{d}.$$

□

We have the following:

Theorem 5 *Suppose that $\mathcal{A}_i$ is an α-approximation algorithm for $\max_{X\in\mathcal{C}} f_i(X)$ with time complexity $T_i(|E|)$ for $i \in [d]$. Then, Algorithm 1 outputs a family of d solutions with regret ratio at most $1 - \alpha/d$ in $O(d + \sum_{i\in[d]} T_i(|E|))$ time.*



(a) Beginning of algorithm (b) After the first loop (c) Approximate case

Figure 1: Investigation of faces by Polytope Algorithm. The x-axis and y-axis represent the values of f_1 and f_2, respectively, and a solution X is identified with a point $\boldsymbol{f}(X)$. (a) The Coordinate-wise maximum method yields points A and B. Then we know that other Pareto points must be below of C. The initial uncovered region is $\triangle ABC$. (b) Next, the algorithm picks a normal vector of face AB and finds point Q. Then we know that other Pareto points are also below of line $B'A'$, which is a line passing through Q and parallel to face AB. Now the uncovered region shrinks into $\triangle A'AQ$ and $\triangle B'BQ$. (c) For the approximate case, one can run a similar argument, but the definition of the uncovered region is changed.

PROOF: The regret ratio is immediate from Lemma 4. The time complexity follows as we run the algorithm $\mathcal{A}_i$ for $i \in [d]$ and the output set S_{coord} has size d. □

4.2 Polytope method

Our second algorithm is based on the geometric characterization of the regret ratio. The algorithm first runs Algorithm 1 to obtain a polytope $P(\mathcal{S})$. For each frontier face F of $P(\mathcal{S})$, we compute a nonnegative normal vector $\boldsymbol{a}$ of F. Note that one can always find a nonnegative normal vector $\boldsymbol{a}$ from the definition of $P(\mathcal{S})$. Then, we run an approximation algorithm for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$ to obtain an approximate solution X, and add X to $\mathcal{S}$. A pseudocode description is presented in Algorithm 2.

To explain the intuitive concept underlying this algorithm, let us consider the case of $d = 2$. An illustration of the algorithm is shown in Figure 1. In the figure, we identify a solution X with a point $\boldsymbol{f}(X)$. The algorithm tries to reduce the area of the region that may contain points not included by $P(\mathcal{S})$, which is shown as the shaded region in Figure 1. Intuitively, the shaded region can be shrunk by taking a normal vector of the face and adding a point maximizing $f_{\boldsymbol{a}}(X)$.

Before analyzing the regret ratio of Algorithm 2, we analyze its time complexity:

Theorem 6 *Suppose $\mathcal{A}$ is an approximation algorithm with time complexity $T(|E|)$. Then, Algorithm 2 runs in $O(k \log k + k^{\lfloor d/2 \rfloor} + (d+k)T(|E|))$ time.*

PROOF: Through the algorithm, the number of invocations of $\mathcal{A}$ is $O(d+k)$. The process of maintaining the faces is essentially equivalent to the dynamic update of a convex hull in d-dimensional space. As we end with adding k points, we can maintain the faces in $O(k \log k + k^{\lfloor d/2 \rfloor})$ time by using the algorithm by [2]. Summing up these time complexities, we get the desired result. □

4.2.1 Analysis for the exact case

First, we analyze Algorithm 2 when $d = 2$, and we can find *exact* solutions for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$. Indeed, our algorithm is closely related to the *Chord* algorithm for approximating convex curves (see [3] and the references therein).

Algorithm 2 Polytope method

Require: Submodular functions $f_1, \dots, f_d : 2^E \to \mathbb{R}_+$, a constraint $\mathcal{C} \subseteq 2^E$, an integer $k \in \mathbb{N}$, and an approximation algorithm $\mathcal{A}$ for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$ ($\boldsymbol{a} \in \mathbb{R}^d_+$).

1: **for** $i \in [d]$ **do**
2: $X_i \leftarrow$ a solution obtained by applying $\mathcal{A}$ to f_i.
3: $\mathcal{S} \leftarrow \{X_1, \dots, X_d\}$, $P \leftarrow P(\mathcal{S})$.
4: **while** $|\mathcal{S}| < k$ **do**
5: **for** each frontier face F of P **do**
6: Find a nonnegative normal vector $\boldsymbol{a}$ of F.
7: $X \leftarrow$ a set obtained by applying $\mathcal{A}$ to $f_{\boldsymbol{a}}$.
8: Add X to $\mathcal{S}$.
9: **if** $|\mathcal{S}| = k$ **then return** $\mathcal{S}$.
10: $P \leftarrow P(\mathcal{S})$.
11: **return** $\mathcal{S}$.

Theorem 7 *Assume that $d = 2$ and that we can find exact solutions for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$. After Algorithm 2 investigates all the faces of P i times, the maximum regret ratio $\mathrm{rr}_{\mathcal{C}}(\mathcal{S})$ is at most $\sqrt{2} \cdot 2^{-i}$.*

For the proof, we analyze the area of the region that may contain points not included by $P(\mathcal{S})$. We refer to this region as the *uncovered region*. For example, in Figure 1, the uncovered regions are represented by the shaded regions. Intuitively, in each iteration, the areas of the uncovered regions shrink. Indeed, the areas shrink exponentially.

Lemma 8 ([3, Lemma 3.11], restated in our context.) *Suppose that Algorithm 2 processes face AB. Let $T = \triangle ABC$ be the part of the uncovered region corresponding to face AB. Denote $Q = \boldsymbol{f}(X)$, where X is a solution found in Line 7. Let $T_1 = \triangle AA'Q$ and $T_2 = \triangle BB'Q$ be the parts of the new covered region corresponding to faces AQ and BQ, respectively. Then, we have $S(T_1) + S(T_2) \le S(T)/4$, where $S(T)$ denotes the area of T.*

Proof:[of Theorem 7] Since the regret ratio is scale-invariant, we can assume that $\max_{X \in \mathcal{C}} f_1(X) = \max_{X \in \mathcal{C}} f_2(X) = \sqrt{2}$. Then, the distance from the origin to any face of $P(\mathcal{S})$ is at least 1, and the area of the initial uncovered region is at most 1. By Lemma 8, after Algorithm 2 processes all the faces of P, the areas of the uncovered regions shrink by a factor of $1/4$. Let us focus on a single triangle $\triangle$ in the uncovered region, and let $h_\triangle$ be the maximum distance from the face to a point in the uncovered region (see Figure 1b). Since $\triangle$ is an obtuse triangle, we have $S(\triangle) \ge h_\triangle^2/2$. Then, $\max_\triangle h_\triangle^2 \le 2 \sum_\triangle S(\triangle) = 2S(\text{uncovered region}) \le 2 \cdot 4^{-i}$. Thus, $\max_\triangle h_\triangle \le \sqrt{2} \cdot 2^{-i}$. By Lemma 3,

$$\begin{aligned}
\mathrm{rr}_{\mathcal{C}}(\mathcal{S}) &= \max_\triangle \frac{h_\triangle}{\mathrm{dist}(\triangle, \mathbf{0}) + h_\triangle} \\
&\le \max_\triangle \frac{h_\triangle}{1 + h_\triangle} \\
&\le \max_\triangle h_\triangle \\
&\le \sqrt{2} \cdot 2^{-i},
\end{aligned}$$

where the first inequality follows from the fact that the distance from the origin to any triangle is at least 1. □

Corollary 9 *After Algorithm 2 adds k solutions to $\mathcal{S}$, $\mathrm{rr}_{\mathcal{C}}(\mathcal{S})$ is at most $\sqrt{2} \cdot 2^{-\lfloor \log_2(k-1) \rfloor} = O(1/k)$.*

Proof: One can check that after Algorithm 2 examines all the faces i times, the number of faces in $P(\mathcal{S})$ is at most $2^i + 1$. Thus, we have $k \le 2^i + 1$, which yields $i \le \lfloor \log_2(k-1) \rfloor$. □

4.2.2 Analysis for the approximate case

Let us analyze the case where we have only an α-approximation algorithm for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$. In this case, the best one can hope for is that $P(\mathcal{C}) \subseteq \alpha^{-1} P(\mathcal{S})$, i.e., any Pareto optimal point is within the α-multiplicative factor.

Theorem 10 *Assume that $d = 2$ and that we can find α-approximate solutions for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$. After Algorithm 2 investigates all the faces of P i times, the maximum regret ratio $\mathrm{rr}_{\mathcal{C}}(\mathcal{S})$ is at most $1 - \alpha + \sqrt{2} \cdot 2^{-i}$.*

Proof: The proof idea is showing that $P(\mathcal{C}) \subseteq (\alpha - \epsilon)^{-1} P(\mathcal{S})$, where ϵ decreases exponentially in i. Let us call the area of the region that may contain points not included by $\alpha^{-1} P(\mathcal{S})$ the *uncovered region* (see Figure 1c). It suffices to show the theorem for the case in which the approximation algorithm for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$ always returns α-approximate solutions. To see this, suppose that we obtain a β-approximate solution ($\beta > \alpha$) for some normal vector $\boldsymbol{a}$ of a face of $P(\mathcal{S})$. Adding this approximate solution to $\mathcal{S}$ reduces the uncovered area more than adding an α-approximate solution. Thus, the analysis reduces to that of the exact case and the theorem follows from Theorem 7. □

The above argument heavily relies on Lemma 8, which is shown only for the two-dimensional case. In higher dimension, the uncovered region becomes complicated; therefore the analysis becomes more difficult. We leave the analysis in higher dimension for future work.

4.3 Reduction Method

In this section, we present an algorithm that achieves almost optimal regret ratio in polynomial time. Note that the running time of the algorithm can be huge in practice and therefore this algorithm is only of theoretical interest. Our algorithm splits into two parts. Assume that we have an α-approximation algorithm for single objective submodular maximization. The first part finds a points set $P \subseteq \mathbb{R}_+^d$ of polynomial size such that any Pareto optimal point is dominated by some point in $\mathrm{conv}(P)$ up to $\alpha(1-\epsilon)$-multiplicative factor, where ϵ is a parameter. Such a set P is called a $\alpha(1-\epsilon)$-*convex set* and can be efficiently computed by an algorithm of [4] in $O(d^{d+1}(\log \Delta/\epsilon)^{d-1})$ calls to single objective submodular maximization, where Δ is the minimum value such that $2^{-\Delta} \leq f_i(X) \leq 2^{\Delta}$ for $i \in [d]$ and $X \subseteq 2^E$. Then, our algorithm passes P to the algorithm [1] to find a regret minimizing family. Since P is a subset in $\mathbb{R}_+^d$ of size polynomial, thus the entire algorithm runs in polynomial. The details are shown in Algorithm 3.

Algorithm 3 Reduction method

Require: Submodular functions $f_1, \ldots, f_d : 2^E \to \mathbb{R}_+$, a constraint $\mathcal{C} \subseteq 2^E$, and an α-approximation algorithm $\mathcal{A}$ for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$ ($\boldsymbol{a} \in \mathbb{R}_+^d$), $k \in \mathbb{Z}_+$, $\Delta > 0$, and $\epsilon > 0$.

Ensure: $\mathcal{S} \subseteq \mathcal{C}$ with $|\mathcal{S}| = k$.

1: $P \leftarrow$ ConvexParetoSet$(f_1, \ldots, f_d, \mathcal{C}, \mathcal{A}, \Delta, \epsilon)$.
2: Pass P to the algorithm of [1] to obtain $S \subseteq P$ of size k.
3: **return** the family $\mathcal{S}$ corresponding to S.

Theorem 11 *For any $\epsilon > 0$, Algorithm 3 finds a family $\mathcal{S} \subseteq \mathcal{C}$ with $|\mathcal{S}| = k$ such that*

$$\mathrm{rr}_{\mathcal{C}}(\mathcal{S}) \leq 1 - \alpha(1-\epsilon)\left[1 - \Omega\left(\frac{1}{k^{2/(d-1)}}\right)\right].$$

Algorithm 3 invokes $O(d^{d+1}(\Delta/\epsilon)^{d-1})$ calls to the algorithm $\mathcal{A}$.

Algorithm 4 ConvexParetoSet$(f_1, \dots, f_d, \mathcal{C}, \mathcal{A}, \Delta, \epsilon)$ (adapted from [3])

Require: Submodular functions $f_1, \dots, f_d : 2^E \to \mathbb{R}_+$, a constraint $\mathcal{C} \subseteq 2^E$, and an α-approximation algorithm $\mathcal{A}$ for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$ $(\boldsymbol{a} \in \mathbb{R}^d_+)$, $\Delta > 0$, and $\epsilon > 0$.

Ensure: $P \subseteq \mathbb{R}^d_+$

Let $R := \emptyset$, $A := \emptyset$, and $M := \lfloor \frac{2(d-1)}{\epsilon} \rfloor$.
for $I = 1, \dots, d$ **do**
  $R := R \cup \{(r_1, \dots, r_d) \in \mathbb{R}^d_+ : r_i = 1 \text{ and } r_j \in \{1, 2, \dots, 2^{2\Delta - 1}\} (j \neq i)\}$.
  $A := A \cup \{(a_1, \dots, a_d) \in \mathbb{R}^d_+ : a_i = 1 \text{ and } a_j \in \{1/M, 2/M, \dots, 1\} (j \neq i)\}$.
for $(r_1, \dots, r_d) \in R$ **do**
  Let $g_i := r_i f_i$ $(i \in [d])$.
  for $\boldsymbol{a} \in A$ **do**
    Let X be an output of $\mathcal{A}$ for $\max_{X \in \mathcal{C}} g_{\boldsymbol{a}}(X)$.
    Add $\boldsymbol{f}(X)$ to P.
return P.

Proof: Let $S := \{\boldsymbol{f}(X) : X \in \mathcal{S}\}$. In [1], it is shown that S satisfies

$$\max_{\boldsymbol{a} \in \mathbb{R}^d_+} \frac{\max_{\boldsymbol{p} \in S} \boldsymbol{a}^\top \boldsymbol{p}}{\max_{\boldsymbol{p} \in P} \boldsymbol{a}^\top \boldsymbol{p}} \geq 1 - \Omega\left(\frac{1}{k^{2/(d-1)}}\right).$$

Evidently $\max_{\boldsymbol{p} \in S} \boldsymbol{a}^\top \boldsymbol{p} = \max_{X \in \mathcal{S}} f_{\boldsymbol{a}}(X)$. Since P is an $\alpha(1-\epsilon)$-convex Pareto set, $\max_{\boldsymbol{p} \in P} \boldsymbol{a}^\top \boldsymbol{p} \geq \alpha(1-\epsilon) \cdot \max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$. Therefore,

$$\max_{\boldsymbol{a} \in \mathbb{R}^d_+} \frac{\max_{X \in \mathcal{S} f_{\boldsymbol{a}}(X)}}{\max_{X \in \mathcal{C} f_{\boldsymbol{a}}(X)}} \geq \alpha(1-\epsilon)\left[1 - \Omega\left(\frac{1}{k^{2/(d-1)}}\right)\right],$$

which completes the proof. The number of calls to $\mathcal{A}$ follows immediately from the algorithm. □

Remark 12 *The above theorem shows that* $\mathrm{rr}_{\mathcal{C}} \leq (1-\alpha)(1-\epsilon) + O(1/k^{\frac{2}{d-1}})$*, which especially yields* $\mathrm{rr}_{\mathcal{C}} \leq O(1/k^2)$ *for* $d = 2$ *and* $\alpha = 1$.

5 Lower Bound

In this section, we show that the trade-off achieved by Algorithm 3 cannot be improved in the two-dimensional case. More specifically, we show the following:

Theorem 13 *For any* k*, there exist* n*,* $f_1, f_2 : 2^E \to \mathbb{R}_+$ *with* $|E| = n$*, and* $\mathcal{C} \subseteq 2^E$ *such that an arbitrary subfamily* $\mathcal{S} \subseteq \mathcal{C}$ *of size* k *has a maximum regret ratio* $\Omega(\frac{1}{k^2})$*.*

Proof: Our construction is inspired by [13, Theorem 4]. Let $f_1(X) := \cos(\frac{\pi |X|}{2n})$ and $f_2(X) := \sin(\frac{\pi |X|}{2n})$. Note that f_1 and f_2 are submodular because $\sin(\frac{\pi x}{2})$ and $\cos(\frac{\pi x}{2})$ are concave for $x \in [0, 1]$. We define $\mathcal{C} := 2^E$, i.e., we do not impose constraints. Let us take an arbitrary $\mathcal{S} \subseteq 2^E$ with $|\mathcal{S}| \leq k$. Without loss of generality, we can assume that two arbitrary distinct elements have different cardinalities (otherwise, we delete some element from $\mathcal{S}$ without losing the regret ratio). We sort the k elements in $\mathcal{S}$ such that $|X_1| < |X_2| < \cdots < |X_k|$. Further, we define $X_0 := \emptyset$ and $X_{k+1} = E$. Let $\phi_i := \frac{\pi |X_i|}{2n}$ $(i = 0, \dots, k+1)$ and define $\theta_i = \phi_i - \phi_{i-1}$ $(i = 1, \dots, k+1)$. Since $\theta_1 + \cdots + \theta_{k+1} = \frac{\pi}{2}$, there exists j such that $\theta_j \geq \frac{\pi}{2(k+1)}$. Define $\beta := \theta_j$. By taking n large enough, we can find $X \subseteq E$ such that $\frac{\pi |X|}{2n} = \phi_j + \frac{\beta}{2} =: \gamma$. Let us consider $\boldsymbol{a} = [\cos\gamma, \sin\gamma]^\top$. One can check that $\max_{X \in 2^E} f_{\boldsymbol{a}}(X) = 1$ and $\max_{X \in \mathcal{S}} f_{\boldsymbol{a}}(X) = f_{\boldsymbol{a}}(X_j) = \cos\gamma\cos\phi_j + \sin\gamma\sin\phi_j = \cos(\frac{\beta}{2})$. Therefore, the regret ratio is $1 - \cos(\frac{\beta}{2}) = \Omega(\frac{\beta^2}{4}) = \Omega(\frac{1}{k^2})$. □

We note that our proof is information theoretic and that it does not rely on any assumption on computational complexity such as P ≠ NP.

References

[1] P. K. Agarwal, N. Kumar, S. Sintos, and S. Suri, "Efficient algorithms for k-regret minimizing sets," *Arxiv*, pp. 1–20, 2017.

[2] K. L. Clarkson and P. W. Shor, "Applications of random sampling in computational geometry, II," *Discrete & Computational Geometry*, vol. 4, no. 5, pp. 387–421, Oct. 1989.

[3] C. Daskalakis, I. Diakonikolas, and M. Yannakakis, "How good is the chord algorithm?" in *SODA*, 2010, pp. 978–991.

[4] I. Diakonikolas, "Approximation of multiobjective optimization problems," Ph.D. dissertation, Columbia University, 2011.

[5] C. W. Ko, J. Lee, and M. Queyranne, "An exact algorithm for maximum entropy sampling," *Operations Research*, pp. 684–691, 1995.

[6] A. Krause, H. B. McMahon, C. Guestrin, and A. Gupta, "Robust submodular observation selection," *Journal of Machine Learning Research*, vol. 9, pp. 2761–2801, 2008.

[7] A. Krause and E. Horvitz, "A utility-theoretic approach to privacy and personalization," in *AAAI*, 2008, pp. 1181–1188.

[8] A. Krause, A. Roper, and D. Golovin, "Randomized sensing in adversarial environments," in *IJCAI*, 2011, pp. 2133–2139.

[9] A. Krause, A. P. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, pp. 235–284, 2008.

[10] H. Lin and J. Bilmes, "Multi-document summarization via budgeted maximization of submodular functions," in *NAACL-HLT*, 2010, pp. 912–920.

[11] ——, "A class of submodular functions for document summarization," in *ACL-HLT*, 2011, pp. 510–520.

[12] X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang, "Selecting stars: The k most representative skyline operator," in *ICDE*, 2007, pp. 86–95.

[13] D. Nanongkai, A. D. Sarma, A. Lall, R. J. Lipton, and J. Xu, "Regret-minimizing representative databases," in *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2. VLDB Endowment, 2010, pp. 1114–1124.

[14] P. Peng and R. C.-W. Wong, "Geometry approach for k-regret query," in *ICDE*, 2014, pp. 772–783.

[15] T. Soma, N. Kakimura, K. Inaba, and K. Kawarabayashi, "Optimal budget allocation: Theoretical guarantee and efficient algorithm," in *ICML*, 2014, pp. 351–359.

[16] Y. Tao, L. Ding, X. Lin, and J. Pei, "Distance-based representative skyline," in *ICDE*, 2009, pp. 892–903.

[17] T. Xia, D. Zhang, and Y. Tao, "On skylining with flexible dominance relation," in *ICDE*, 2008, pp. 1397–1399.

[18] M. L. Yiu and N. Mamoulis, "Multi-dimensional top-k dominating queries," *The VLDB Journal*, vol. 18, no. 3, pp. 695–718, 2009.

[19] Y. Yue and C. Guestrin, "Linear submodular bandits and their application to diversified retrieval," in *NIPS*, 2011, pp. 2483–2491.

An asymptotically improved upper bound on the diameter of polyhedra

Noriyoshi Sukegawa*

Dept. Information and System Engineering
Chuo University
1-13-27 Kasuga, Bunkyo-ku, 112-8551, Japan
sukegawa@ise.chuo-u.ac.jp

Abstract: Kalai and Kleitman proved in 1992 that the maximum possible diameter of a d-dimensional polyhedron with n facets is at most $n^{2+\log_2 d}$. In 2014, Todd improved the Kalai-Kleitman bound to $(n-d)^{\log_2 d}$. Todd's bound is tight for $d \leq 2$, and has been improved for $d \geq 3$ by the subsequent studies. The current best upper bound is, however, still in the form $(n-d)^{\log_2(d/\alpha)}$, where α is some fixed positive constant. This paper shows an asymptotically improved upper bound of $(n-d)^{\log_2 \mathrm{O}(d/\log d)}$.

Keywords: Polyhedra, Diameter, Polynomial Hirsch conjecture

1 Introduction

The diameter $\delta(P)$ of a polyhedron P is the smallest integer k such that every pair of vertices of P can be connected by a path using at most k edges of P. The diameter is closely related to the complexity of the simplex algorithm for solving linear programming problems. Specifically, the number of pivots required, in the worst case, for the simplex algorithm to solve an instance on a polyhedron P is bounded from below by its diameter $\delta(P)$.

This paper is concerned with the maximum possible diameter $\Delta(d,n)$ of a d-dimensional polyhedron with n facets. Note that $\Delta(d,n)$ corresponds to the worst-case complexity of the simplex algorithm to solve an instance with d variables and n constraints. Using this notation, the Hirsch conjecture posed by Warren M. Hirsch in 1957 can be restated as

$$\Delta(d,n) \leq n-d.$$

The Hirsch conjecture is true for $d \leq 3$ as demonstrated by Klee [6, 7, 8]. However, it is now known to be false for $d \geq 4$ in general as demonstrated by Klee and Walkup [9] in 1967 for unbounded polyhedra, and by Santos [11], finally, in 2010 even for bounded polyhedra, i.e., for polytopes. For the history of the Hirsch conjecture, see Santos [12].

It is, however, still wide open what is a *good* upper bound on $\Delta(d,n)$. The current best lower bound on $\Delta(d,n)$, which is due to Santos, later refined by Matschke, Santos, and Weibel [10], violates Hirsch's bound of $n-d$ by only 25%. On the other hand, the current best upper bound is only subexponential in d and n, and is in the form $(n-d)^{\log_2(d/\alpha)}$, where α is some fixed positive constant as demonstrated by Todd [15] in 2014, and by the subsequent studies [13, 14].

In view of the significant gap between the upper and lower bounds on $\Delta(d,n)$, the existence of a polynomial upper bound $p(d,n)$, which is referred to as *the polynomial Hirsch conjecture*, has become a major question. The polynomial Hirsch conjecture is, of course, concerned with the polynomiality of the simplex algorithm for solving linear programming problems, i.e., if the polynomial Hirsch conjecture is false, then the simplex algorithm might not be a polynomial-time algorithm.

*Research is supported by JSPS KAKENHI Grant Number 15H06617.

1.1 Main result

In view of the current best upper bound having the form $(n-d)^{\log_2(d/\alpha)}$, where α is some fixed positive constant, it would be natural to ask whether

$$\Delta(d,n) \leq (n-d)^{\log_2 g(d)}$$

holds for some function $g(d)$ which is *asymptotically* smaller than d, to *approach* the polynomiality. The aim of this paper is to show that the answer to the question above is "yes":

Theorem 1 *For $n \geq d \geq 2$,*

$$\Delta(d,n) \leq (n-d)^{\log_2 g(d)},$$

where $g(d) = 3h(d)$ with

$$h(d) = \sum_{k=2}^{d} \frac{1}{\log_2 k}.$$

Observation 2 *Using the sum-integral argument, it is easily seen that*

$$\begin{aligned} h(d) = \sum_{k=2}^{d} \frac{1}{\log_2 k} &\leq 1 + \int_2^d \frac{dt}{\log_2 t} \\ &= 1 + \ln 2 \int_2^d \frac{dt}{\ln t} \\ &= 1 + \ln 2 \cdot \mathrm{Li}(d), \end{aligned}$$

where $\mathrm{Li}(d)$ denotes the offset logarithmic integral, which is well known to be $\mathrm{O}(d/\log d)$ for $d \geq 2$. We give an alternative proof to this fact that $h(d) = \mathrm{O}(d/\log d)$, as well as $h(d) = \Omega(d/\log d)$ in Appendix A. These imply $g(d) = \Theta(d/\log d)$.

Corollary 3 *For $n \geq d \geq 2$,*

$$\Delta(d,n) \leq (n-d)^{\log_2 \mathrm{O}(d/\log d)}.$$

Figure 1 shows the comparison results on the values of Todd's bound, i.e., $(n-d)^{\log_2 d}$ (dotted) and ours, i.e., $(n-d)^{\log_2 g(d)}$ (solid). Figure 1 (a) shows those when n is fixed to $2d$ and d ranges from 2 to 50. On the other hand, Figure 1 (b) shows those when $d = 32$ and $n-d$ ranges from 0 to 100.

1.2 Several remarks on Theorem 1

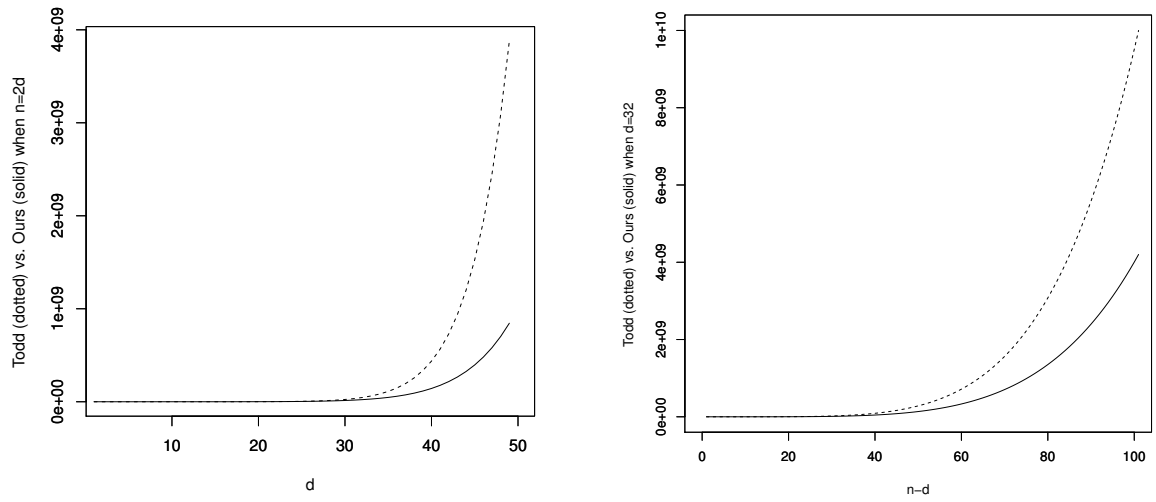
1.2.1 Motivation

The upper bounds on $\Delta(d,n)$ shown in [13, 14, 15] are all derived from the same recursive inequality (restated as Lemma 6 in this paper) established by Kalai and Kleitman [4], who used it in 1992 to show the *first* subexponential, or quasi polynomial, upper bound of $n^{2+\log_2 d}$. Our proof also makes use of the same recursive inequality, and is inspired by the proof and observations in [13]. It was shown in [13] that

a) $\Delta(d,n) \leq (n-d)^{\log_2(d/2)}$ for $d \geq 7$, and

b) $\Delta(d,n) \leq (n-d)^{\log_2(d/4)}$ for $d \geq 37$.

These upper bounds might imply that the answer to the question mentioned earlier is "yes". This paper introduces a novel alternative approach to the recursive inequality, and consequently proves Theorem 1. See Remark 9 for more formal discussions on the difference from the previous proof techniques used in [13, 14, 15].



(a) When $n = 2d$ (b) When $d = 32$

Figure 1: Values of Todd's bound (dotted) and ours (solid)

1.2.2 How to further refine Theorem 1

In [13], the upper bounds on $\Delta(d, n)$ were shown by induction on d. The key was the observation that its inductive step can be *tighten* in high dimensions. If one wants to apply the inductive step for $d \geq k$, then the validity of the target upper bound must be verified in dimension $d = k - 1$ as the base case. In order to deal with this base case, in [13], a computer-assisted tool was devised.

Our proof is also by induction on d, and hence applying the computer-assisted tool of [13] to our analysis can reduce the constant factor hidden in the estimation of $\mathrm{O}(d/\log d)$. However, since this makes the discussions complicated, and is out of the scope of this paper, we omit the details.

1.2.3 Extension to abstractions

The upper bound stated in Theorem 1 may hold in several abstractions, i.e., objects generalizing polyhedra. This follows from the observations in Gallagher and Kim [3], where they demonstrated that the diameter bound for polyhedra shown in [14] can be easily generalized to that for normal simplicial complexes. This is because the key ingredient, the recursive inequality stated in Lemma 6, also applies to normal simplicial complexes.

For notable lower and upper bounds on the diameter of abstractions, we refer to [1, 2, 5] and the references therein.

1.3 Terminologies

We first fix some definitions and notations. A polyhedron P is an intersection of a finite number of closed halfspaces. Here, the polyhedron can be unbounded. The dimension $\dim(P)$ of P is defined as the dimension of its affine hull. For a polyhedron P, an inequality $a^\top x \leq \beta$ is said to be valid for P if it is satisfied by every $x \in P$. We say that F is a face of P if there exists a valid inequality $a^\top x \leq \beta$ for P satisfying $F = P \cap \{x \in \mathbb{R}^d : a^\top x = \beta\}$. In particular, 0-, 1-, and $(\dim(P) - 1)$-dimensional faces are, respectively, referred to as vertices, edges, and facets.

The diameter $\delta(P)$ of a polyhedron P is the smallest integer k such that every pair of vertices of P

can be connected by a path using at most k edges of P. In this paper, we are interested in bounding

$$\Delta(d,n) = \max\{\delta(P) : P \text{ is a } d\text{-dimensional polyhedron with } n \text{ facets}\}$$

from above by a function of d and n. It is always assumed that $n \geq d$. If otherwise, there exists no vertex, and hence $\Delta(d,n) = 0$.

2 Proof of Theorem 1

We use induction on d to prove the upper bound stated in Theorem 1.

2.1 The base case

As we will see later, the inductive step applies only for $d \geq 8$. In other words, as the base case, we need to show that our bound is valid for $2 \leq d \leq 7$. This is verified in the following lemma.

Lemma 4 *For $n \geq d$ with $2 \leq d \leq 7$, our bound is valid, i.e., we have $\Delta(d,n) \leq (n-d)^{\log_2 g(d)}$.*

PROOF: Recalling that $g(d) = 3h(d) = 3\sum_{k=2}^{d}(\log_2 k)^{-1}$, it is easy to observe that $g(d) \geq d$ for $d = 2, 3$. On the other hand, for $4 \leq d \leq 7$,

$$\begin{aligned} g(d) = 3h(d) = 3\sum_{k=2}^{d}\frac{1}{\log_2 k} &\geq 3\left[1 + \frac{1}{2} + \sum_{k=4}^{d}\frac{1}{\log_2 k}\right] \\ &\geq 3\left[1 + \frac{1}{2} + \frac{d-3}{\log_2 d}\right] \\ &\geq 3\left[1 + \frac{1}{2} + \frac{d-3}{3}\right] \\ &\geq d. \end{aligned}$$

Since Todd's bound is $(n-d)^{\log_2 d}$, and it is a valid upper bound on $\Delta(d,n)$ for $n \geq d$ with any d, we see that for $n \geq d$ with $2 \leq d \leq 7$,

$$\Delta(d,n) \leq (n-d)^{\log_2 d} \leq (n-d)^{\log_2 g(d)}.$$

This completes the proof of the lemma. □

2.2 The inductive step

Let d be an integer such that $d \geq 8$. Assume that our bound is valid in dimension $d-1$. We use induction on n to prove the inequality $\Delta(d,n) \leq (n-d)^{\log_2 g(d)}$ for each n such that $n \geq d$.

2.2.1 When $n < 2d$ (base case):

It is known that:

Proposition 5 (e.g., Klee and Walkup [9]) *For $n < 2d$, $\Delta(d,n) \leq \Delta(d-1, n-1)$.*

From Proposition 5 and the induction hypothesis, in this case,

$$\Delta(d,n) \leq \Delta(d-1,n-1) \leq (n-d)^{\log_2 g(d-1)} \leq (n-d)^{\log_2 g(d)}$$

where the last inequality follows since $g(d)$ is an increasing function of d, and $n-d$ is a nonnegative integer.

2.2.2 When $n \geq 2d$ (inductive step):

Let n be an integer such that $n \geq 2d$. Assume as the induction hypothesis on n, that our bound is valid in dimension d if the number of facets is smaller than n. We now make use of the following lemma established by Kalai and Kleitman in 1992:

Lemma 6 (Kalai and Kleitman [4]) *For* $\lfloor n/2 \rfloor \geq d \geq 2$,

$$\Delta(d,n) \leq \Delta(d-1,n-1) + 2\Delta\left(d, \left\lfloor \frac{n}{2} \right\rfloor\right) + 2.$$

For convenience, let $f(d,n)$ denote our bound, i.e., $f(d,n) = (n-d)^{\log_2 g(d)}$. Combined with the induction hypotheses on d, as well as on n, Lemma 6 tells us

$$\begin{aligned}\Delta(d,n) &\leq \Delta(d-1,n-1) + 2\Delta\left(d, \left\lfloor \frac{n}{2} \right\rfloor\right) + 2 \\ &\leq f(d-1,n-1) + 2f\left(d, \left\lfloor \frac{n}{2} \right\rfloor\right) + 2.\end{aligned}$$

Therefore, it suffices to show

$$f(d-1,n-1) + 2f\left(d, \left\lfloor \frac{n}{2} \right\rfloor\right) + 2 \leq f(d,n).$$

Since $f(d,n) > 0$ for $n \geq 2d$ with $d \geq 8$, it can be rewritten as

$$\frac{f(d-1,n-1)}{f(d,n)} + 2\frac{f\left(d, \lfloor n/2 \rfloor\right)}{f(d,n)} + \frac{2}{f(d,n)} \leq 1. \tag{1}$$

Note that in general, $a^{\log_2 b} = b^{\log_2 a}$ for $a, b > 0$. Hence, for $n \geq 2d$ with $d \geq 8$,

$$f(d,n) = (n-d)^{\log_2 g(d)} = g(d)^{\log_2 (n-d)}.$$

Using the last expression $g(d)^{\log_2(n-d)}$, LHS of (1) is

$$\left[\frac{g(d-1)}{g(d)}\right]^{\log_2(n-d)} + \frac{2g(d)^{\log_2(\lfloor n/2 \rfloor - d)}}{g(d)^{\log_2(n-d)}} + \frac{2}{g(d)^{\log_2(n-d)}}.$$

Noting that $\lfloor n/2 \rfloor - d \leq (n-d)/2$, it is bounded from above by

$$\begin{aligned}&\left[\frac{g(d-1)}{g(d)}\right]^{\log_2(n-d)} + \frac{2g(d)^{\log_2((n-d)/2)}}{g(d)^{\log_2(n-d)}} + \frac{2}{g(d)^{\log_2(n-d)}} \\ &= \left[\frac{g(d-1)}{g(d)}\right]^{\log_2(n-d)} + \frac{2g(d)^{-1+\log_2(n-d)}}{g(d)^{\log_2(n-d)}} + \frac{2}{g(d)^{\log_2(n-d)}} \\ &= \left[\frac{g(d-1)}{g(d)}\right]^{\log_2(n-d)} + \frac{2}{g(d)} + \frac{2}{g(d)^{\log_2(n-d)}}.\end{aligned} \tag{2}$$

It follows from the definition of $g(d)$ that $g(d-1)/g(d) < 1$ and $g(d) > 1$ for $d \geq 8$. Then, recalling that we now assume $n \geq 2d$ implying $n - d \geq d$, (2) can be bounded from above by

$$\left[\frac{g(d-1)}{g(d)}\right]^{\log_2(d)} + \frac{2}{g(d)} + \frac{2}{g(d)^{\log_2(d)}}. \tag{3}$$

Now, observe that for $d \geq 8$,

$$\begin{aligned} h(d) = \sum_{k=2}^{d} \frac{1}{\log_2 k} &\geq \sum_{k=2}^{7} \frac{1}{\log_2 k} = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5} + \frac{1}{\log_2 6} + \frac{1}{\log_2 7} \\ &\geq \frac{1}{\log_2 2} + \frac{1}{\log_2 4} + \frac{1}{\log_2 4} + \frac{1}{\log_2 8} + \frac{1}{\log_2 8} + \frac{1}{\log_2 8} \\ &= 1 + 2 \cdot \frac{1}{2} + 3 \cdot \frac{1}{3} \\ &= 3, \end{aligned}$$

which implies that $g(d) \geq 9$ for $d \geq 8$, because $g(d) = 3h(d)$. Thus, for $d \geq 8$, the third term of (3) is bounded from above as follows:

$$\frac{2}{g(d)^{\log_2(d)}} \leq \frac{2}{g(d)^3} = \frac{2}{g(d)^2} \cdot \frac{1}{g(d)} \leq \frac{2}{81} \cdot \frac{1}{g(d)}.$$

Therefore, it suffices to show the following.

Claim 7 *For $d \geq 8$,*

$$\left[\frac{g(d-1)}{g(d)}\right]^{\log_2(d)} + \left(2 + \frac{2}{81}\right) \frac{1}{g(d)} \leq 1.$$

Proof: It follows from the definition of $g(d)$ that $g(d) - g(d-1) = 3/\log_2 d$, which implies

$$\frac{g(d-1)}{g(d)} = 1 - \frac{3}{g(d) \log_2 d}.$$

We now make use of the following proposition whose proof immediately follows from the fact that $1 + x \leq \exp(x)$ for $x \in \mathbb{R}$:

Proposition 8 *For $x > -n$ with $n > 0$,*

$$\left(1 + \frac{x}{n}\right)^n \leq \exp(x).$$

Observe that setting $n \equiv \log_2 d$ and $x \equiv -3/g(d)$, the conditions required in Proposition 8 are satisfied because for $d \geq 8$,

- $n \equiv \log_2 d \geq \log_2 8 > 1 > 0$, and
- $x \equiv -\dfrac{3}{g(d)} \geq -\dfrac{3}{g(8)} = -\dfrac{3}{9} \geq -1 > -n.$

Then, Proposition 8 tells us that

$$\begin{aligned} \left[\frac{g(d-1)}{g(d)}\right]^{\log_2 d} = \left[1 + \frac{-\frac{3}{g(d)}}{\log_2 d}\right]^{\log_2 d} &\leq \exp\left(-\frac{3}{g(d)}\right) = \frac{1}{\exp\left(\frac{3}{g(d)}\right)} \\ &\leq \frac{1}{1 + \frac{3}{g(d)}} = \frac{g(d)}{g(d)+3} \end{aligned}$$

where the last inequality follows from the fact that $1 + x \leq \exp(x)$ for $x \in \mathbb{R}$. By the discussion so far, it suffices to show

$$\frac{g(d)}{g(d)+3} + \left(2 + \frac{2}{81}\right) \frac{1}{g(d)} \leq 1,$$

which can be simplified to

$$g(d) \geq \frac{3\left(2+\frac{2}{81}\right)}{3-\left(2+\frac{2}{81}\right)}. \tag{4}$$

Note that the right hand side value is less than seven. On the other hand, the proof of Lemma 4 tells us that $g(d) \geq 7$ for $d \geq 8$, which ensures (4) when $d \geq 8$. This completes the proof of the claim. □

To summarize, the inductive step works correctly for $d \geq 8$. Combined with Lemma 4 showing the base case for $2 \leq d \leq 7$, this completes the proof of Theorem 1.

As a final remark on our proof, below, we discuss the differences from the previous proof techniques used in [13, 14, 15].

Remark 9 *In [15], in order to prove the upper bound of $(n-d)^{\log_2 d}$, i.e., the case $g(d) = d$, Todd designed his inductive step so that it applies for $n - d \geq 8$. Since $n - d \geq 8$ implies $\log_2(n-d) \geq 3$, in this case, (2) with $g(d) = d$ is bounded from above by*

$$\left[\frac{d-1}{d}\right]^3 + \frac{2}{d} + \frac{2}{d^3}.$$

This rewriting makes the subsequent analyses somewhat simple. Then, slightly extending this Todd's idea, [13] introduced a framework for proving upper bounds under the assumption that $n - d \geq 2^m$ for some fixed positive integer m.

This paper, in contrast, directly uses the assumption that $n - d \geq d$, which makes the analysis much different from those of [13, 15].

Acknowledgements
The author is grateful to Kazuo Murota and Yoshio Okamoto for stimulating this study. The author is also grateful to Yuya Higashikawa for his helpful comments on the alternative proof of $\mathrm{O}(d/\log d)$ upper bound on $g(d)$, discussed in Appendix A. This work was supported by JSPS KAKENHI Grant Number 15H06617.

References

[1] F. Eisenbrand, N. Hähnle, A. Razborov, and T. Rothvoss: Diameter of polyhedra: limits of abstraction. *Mathematics of Operations Research* 35 (2010) 786–794.

[2] J.M. Gallagher and E.D. Kim: An improved upper bound on the diameters of subset partition graphs. `https://arxiv.org/abs/1412.5691`

[3] J.M. Gallagher and E.D. Kim: Tail diameter upper bounds for polytopes and polyhedra. `https://arxiv.org/abs/1603.04052`

[4] G. Kalai and D.J. Kleitman: A quasi-polynomial bound for the diameter of graphs of polyhedra. *Bulletin of the American Mathematical Society* 26 (1992) 315–316.

[5] E.D. Kim: Polyhedral graph abstractions and an approach to the Linear Hirsch conjecture. *Mathematical Programming, Series A* 143 (2014) 357–370.

[6] V. Klee: Diameters of polyhedral graphs. *Canadian Journal of Mathematics* 16 (1964) 602–614.

[7] V. Klee: Paths on polyhedra: I. *Journal of the Society for Industrial and Applied Mathematics* 13 (1965) 946–956.

[8] V. Klee: Paths on polyhedra: II. *Pacific Journal of Mathematics* 17 (1966) 249–262.

[9] V. Klee and D.W. Walkup: The d-step conjecture for polyhedra of dimension $d < 6$. *Acta Mathematica* 133 (1967) 53–78.

[10] B. Matschke, F. Santos, and C. Weibel: The width of 5-dimensional prismatoids. `http://arxiv.org/abs/1202.4701`.

[11] F. Santos: A counter-example to the Hirsch Conjecture. *Annals of Mathematics* 176 (2012) 383–412.

[12] F. Santos: Recent progress on the combinatorial diameter of polytopes and simplicial complexes. *Top* 21 (2013) 426–460.

[13] N. Sukegawa: Improving bounds on the diameter of a polyhedron in high dimensions. `https://arxiv.org/abs/1604.04039`

[14] N. Sukegawa and T. Kitahara: A refinement of Todd's bound for the diameter of a polyhedron. *Operations Research Letters* 43 (2015) 534–536.

[15] M.J. Todd: An improved Kalai-Kleitman bound for the diameter of a polyhedron. *SIAM Journal on Discrete Mathematics* 28 (2014) 1944–1947.

A Proof of $g(d) = \Theta\left(d/\log d\right)$

It suffices to show that there are positive numbers C_1 and C_2, and integer d^* such that if $d \geq d^*$, then

$$C_1 \frac{d}{\log_2 d} \leq h(d) \leq C_2 \frac{d}{\log_2 d}.$$

Proof of a lower bound: The lower bound immediately follows since for $d \geq 2$,

$$h(d) = \sum_{k=2}^{d} \frac{1}{\log_2 k} = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \cdots + \frac{1}{\log_2 d} \geq (d-1) \cdot \frac{1}{\log_2 d} \geq \frac{1}{2} \cdot \frac{d}{\log_2 d}.$$

Proof of an upper bound: We show that $h(d) \leq 3d/\log_2 d$ for $d \geq 8$ by induction on d. It is easy to see that the base case $d = 8$ is true. Now, observe that for $d \geq 9$,

$$\begin{aligned}
\left[\frac{d}{\log_2 d} - \frac{d-1}{\log_2(d-1)}\right] \log_2 d &= \frac{d\log_2(d-1) - (d-1)\log_2 d}{\log_2(d-1)} \\
&= \frac{\log_2(d-1) + \log_2\left(1-\frac{1}{d}\right)^{d-1}}{\log_2(d-1)} \\
&\geq \frac{\log_2(d-1) + \log_2\left(1-\frac{1}{d-1}\right)^{d-1}}{\log_2(d-1)} \\
&\geq 1 + \frac{\log_2\left(1-\frac{1}{8}\right)^{8}}{\log_2(d-1)} \\
&\geq 1 - 2 \cdot \frac{1}{\log_2(d-1)} \\
&\geq 1 - 2 \cdot \frac{1}{3} \\
&= \frac{1}{3},
\end{aligned}$$

where the second inequality follows from Claim 10 below. Hence, assuming the induction hypothesis,

$$\begin{aligned} h(d) &= h(d-1) + \frac{1}{\log_2 d} \\ &\leq 3 \cdot \frac{d-1}{\log_2(d-1)} + 3\left[\frac{d}{\log_2 d} - \frac{d-1}{\log_2(d-1)}\right] \\ &= 3 \cdot \frac{d}{\log_2(d)}. \end{aligned}$$

Claim 10 *If n is an integer with $n \geq 2$, then $\left(1 - \frac{1}{n+1}\right)^{n+1} \geq \left(1 - \frac{1}{n}\right)^n$*

PROOF: We give a proof for completeness. Observe that for $n \geq 2$,

$$\begin{aligned} \left(1 - \frac{1}{n+1}\right)^{n+1} \Big/ \left(1 - \frac{1}{n}\right)^n &= \left(1 + \frac{1}{n^2-1}\right)^n \left(1 - \frac{1}{n+1}\right) \\ &\geq \left(1 + \frac{n}{n^2-1}\right)\left(1 - \frac{1}{n+1}\right) \\ &\geq 1 \end{aligned}$$

where the first inequality follows from Bernoulli's inequality stating that $(1+x)^n \geq 1 + nx$ for every nonnegative integer n and real number x with $x \geq -1$. □

Three Theorems on the Combinatorics of Finite Metric Spaces

Péter G.N. Szabó[1]

Alfréd Rényi Institute of Mathematics
Hungarian Academy of Sciences
and
Department of Computer Science and
Information Theory
Budapest Univ. of Technology and Economics
3-9., Műegyetem rkp., H-1111 Budapest,
Hungary
szape@cs.bme.hu

Abstract: In this paper we present three theorems from the topic of combinatorics of finite metric spaces. We characterize linear betweenness structures, uniquely representable graphs and we show that the Fano plane is not metrizable.

Keywords: finite metric space, betweenness, Fano plane, Husimi tree

1 Introduction

Metric space is one of the most successful concept of mathematics with various applications in computer science, quantitative geometry, topology and phylogenetics. Although finite metric spaces are trivial objects from a topological point of view, they have surprisingly complex and intriguing combinatorial properties which were investigated from different angles over the last fifty years.

Metric properties of trees were studied by Buneman [3] who introduced the famous four-point condition. That work was continued by Dress et al. who studied both algorithmic and combinatorial aspects of phylogenetic trees and split decompositions of finite metric spaces [2, 6]. A more recent problem of the field is the Chen-Chvátal conjecture [4], which generalizes the De Bruijn-Erdős theorem for finite metric spaces. The conjecture was proven for distance-hereditary graphs by Aboulker and Kapadia [1]. Finally, Mascioni proved novel results on Ramsey numbers of finite metric spaces in [9].

The main focus of this paper is the combinatorics of the betweenness relation on finite metric spaces. In order to give an overall picture on the topic, we answer three interesting questions: we characterize linear betweenness structures, uniquely representable graphs and we show that the Fano plane is not metrizable.

A metric space $M = (X, d)$ is finite if $|X| < \infty$. Every metric space in this paper is assumed to be finite if not stated otherwise. By graph we always mean a simple undirected graph. Let $G = (V, E)$ be a connected graph. The *metric space induced by* G is $M(G) = (V, d_G)$, where $d_G(u, v)$ is the length of the shortest path between u and v in G.

A *betweenness structure* is a pair $\mathcal{B} = (X, \beta)$, where X is a nonempty finite set and $\beta \subseteq X^3$ is a ternary relation, called the *betweenness relation* of $\mathcal{B}$. The fact $(x, y, z) \in \beta$ will be denoted by $(x\ y\ z)_{\mathcal{B}}$ or simply $(x\ y\ z)$ if $\mathcal{B}$ is clear from the context. In that case, we say that y is between x and z or that the 3-set $\{x, y, z\}$ is *collinear* with *middle point* y. Let $\mathcal{B}$ and $\mathcal{C}$ be two betweenness structures over X and Y, respectively. We say that $\mathcal{B}$ and $\mathcal{C}$ are *isomorphic* if there exists a bijection $\varphi : X \to Y$ such that for all $x, y, z \in X$, $(x\ y\ z)_{\mathcal{B}} \Leftrightarrow (\varphi(x)\ \varphi(y)\ \varphi(z))_{\mathcal{C}}$.

[1]Research is supported by the National Research, Development and Innovation Office NKFIH, No. 108947.

There is a natural way to associate a betweennes structure with a metric space. The *betweenness structure induced by a metric space* $M = (X, d)$ is $\mathcal{B}(M) = (X, \beta_M)$, where $\beta_M = \{(x, y, z) \in X^3 : d(x, z) = d(x, y) + d(y, z)\}$ is the *betweenness relation of* M. A betweenness structure is *metrizable* if it is induced by a metric space. In the rest of the paper, every betweenness structure is assumed to be metrizable if not stated otherwise.

The *betweenness structure induced by a connected graph* G is the betweenness structure of the metric space $M(G)$, also denoted by $\mathcal{B}(G)$. A betweenness structure (or metric space) is *graphic* if it is induced by a graph. The *adjacency graph* of a betweenness structure $\mathcal{B} = (X, \beta)$ is $G(\mathcal{B}) = (X, E(\mathcal{B}))$ where $E(\mathcal{B}) = \{\{x, z\} \in \binom{X}{2} : \forall\, y \in X,\ (x\ y\ z)_{\mathcal{B}} \Rightarrow y = x \vee y = z\}$.

2 Main Results

Linearity plays an important role in numerous fields of mathematics. Linear structures include lines in the Euclidean space, paths in graphs and totally ordered sets. The common feature of these structures is that for any three points, one can always pick the one between the other two. We can easily generalize this 3-point condition of linearity to betweenness structures.

Let $\mathcal{B}$ be a betweenness structure over X. A set $Y \subseteq X$ is *collinear* if any three points in Y are collinear according to $\mathcal{B}$. A *line* of $\mathcal{B}$ is a maximal collinear subset of X. We say that $\mathcal{B}$ is *linear* if X is, itself, a line of $\mathcal{B}$ (for a different, less restrictive definition of linearity we refer the reader to [4]). It is then a natural question to ask, what are the linear betweenness structures. Observe that this is an extremal problem in nature as linearity means having the maximal possible number of betweennesses. Let $\mathcal{P}_n$ and $\mathcal{C}_n$ denote the graphic betweenness structures induced by the path and the cycle of length n, respectively. The following theorem characterizes the linear betweenness structures up to isomorphism.

Theorem 1 *A betweenness structure is linear if and only if it is isomorphic to* $\mathcal{C}_4$ *or to* $\mathcal{P}_n$ $(n \geq 1)$.

We remark that the analogue of the Chen-Chvátal conjecture for our definition of linearity is quite straightforward.

Proposition 2 *If* $\mathcal{B}$ *is a nonlinear betweenness structure over* n *points, then* $\mathcal{B}$ *has at least* 3 *lines, and this bound is best possible.*

If G is connected graph, then the adjacency graph of $\mathcal{B}(G)$ is G. Of course, there might be other betweenness structures which satisfy $G(\mathcal{B}) = G$. A graph G is called *uniquely representable* if it is connected and $\mathcal{B}(G)$ is the unique betweenness structure with adjacency graph G. We observed that some important classes of betweenness structures satisfy this property. For example, the following remark of Dress from [5] implies that trees are uniquely representable.

Proposition 3 (Dress [5]) *Let* $\mathcal{B}$ *be a betweenness structure such that* $G(\mathcal{B})$ *is a tree. Then* $\mathcal{B}$ *is induced by a tree.*

A *Husimi tree* [7] is a connected graph in which every cycle induces a complete subgraph. Purely metric characterizations of Husimi trees were given in [7] and [8].

Theorem 4 (Howorka [7]) *A graph is a Husimi tree if and only if it satisfies the four-point condition.*

For the "four-point condition", see [3]. Our next theorem gives a characterization of uniquely representable graphs. It also turns out to be a novel metric characterization of Husimi trees.

Theorem 5 *A graph is uniquely representable if and only if it is a Husimi tree.*

Finally, we take a look at a concrete metrization problem. For a betweenness structure $\mathcal{B} = (X, \beta)$, let $\mathcal{E}(\mathcal{B})$ denote the family of collinear 3-sets of $\mathcal{B}$. Then, $\mathcal{H}(\mathcal{B}) = (X, \mathcal{E}(\mathcal{B}))$ is a 3-uniform hypergraph, called the *collinearity structure* of $\mathcal{B}$. The *collinearity structure induced by a metric space* M is the collinearity structure of $\mathcal{B}(M)$. A collinearity structure is *metrizable* if it is induced by a metric space. It is well known, that the Fano plane (Figure 1) cannot be embedded into the Euclidean plane. We strengthen this result by showing that the Fano plane is not even metrizable.

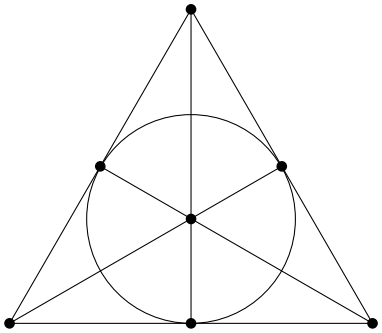


Figure 1: The Fano plane

Theorem 6 *The Fano plane, as a collinearity structure, is not metrizable.*

PROOF: The proof is a thorough case-analysis. We list all possible betweenness structures of the Fano-plane. The number of cases is reduced efficiently by the following lemmas.

Lemma 7 *Suppose that $\mathcal{B}$ is a betweenness structure such that $\mathcal{H}(\mathcal{B})$ is a Steiner triple system. Let $P = (p(x))_{x \in X}$ be a partition of $|\mathcal{E}|$ into $|X|$ parts, where $p(x) = |\{e \in \mathcal{E} : x \text{ is the middle point of } e \text{ in } \mathcal{B}\}|$. Then, the sum of the k greatest parts of P is bounded above by $k(\frac{n-1}{2} - \frac{k-1}{3})$.*

Lemma 8 *The betweenness structures shown in Figure 2 are not metrizable.*

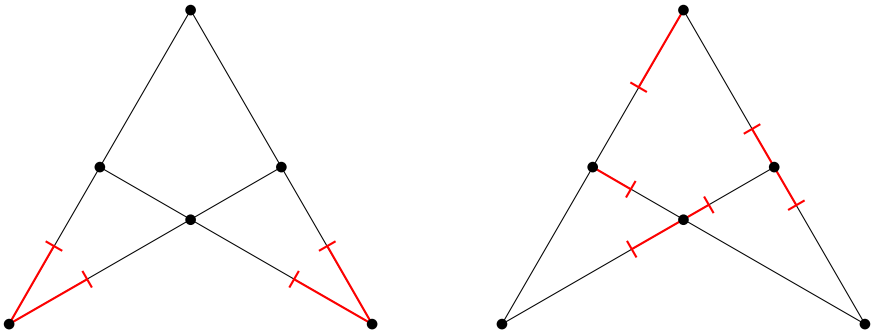


Figure 2: Two non-metrizable betweenness structures on six points. Points on the same line are collinear. The middle point of each line is highlighted by a red line segment.

The final step is to verify that the remaining betweenness structures are non-metrizable. We do this by showing, for each one, a set of strict triangle inequalities that adds up to the contradiction "$0 < 0$". □

3 Open problems

Besides linearity (Theorem 1), other concepts from geometry can also be extended to metric spaces. We call a betweenness structure

- *geometric* if any two distinct lines of it intersect in at most one point;
- *projective* if any two distinct lines of it intersect in exactly one point;

- *Euclidean* if it is embeddable into Euclidean space (in a collinearity-preserving way).

In the graphic case, we found a simple characterization of these classes of betweenness structures. The non-graphic case is still open.

Let $\mathcal{B}$ and $\mathcal{C}$ be two betweenness structures over X. We say that $\mathcal{B}$ is an *extension* of $\mathcal{C}$ ($\mathcal{B} \preccurlyeq \mathcal{C}$) if for all $x, y, z \in X$, $(x\ y\ z)_{\mathcal{C}}$ only if $(x\ y\ z)_{\mathcal{B}}$. One can generalize uniquely representable graphs in the following way. A graph G *bounds its representations from below* if every betweenness structure $\mathcal{B}$ with adjacency graph G satisfies $\mathcal{B}(G) \preccurlyeq \mathcal{B}$. Similarly, G *bounds its representations from above* if every betweenness structure $\mathcal{B}$ with adjacency graph G satisfies $\mathcal{B} \preccurlyeq \mathcal{B}(G)$. Clearly, uniquely representable graphs satisfy these conditions. Further, there are small graphs –for example $K_{2,3}$– that are not uniquely representable but bound their representations from below. However, we conjecture that every graph that bounds its representation from above is uniquely representable. In any case, characterization of these graph classes would generalize Theorem 5 and may give a new perspective on well-known graph classes.

Finally, we believe that Theorem 6 can be extended to Steiner triple systems as well as to finite projective planes, even though our proof works only for the Fano plane. We will further investigate these questions in future research.

References

[1] P. Aboulker, R. Kapadia, The Chen-Chvátal conjecture for metric spaces induced by distance-hereditary graphs, *European J. Combin.* **43** (2015) 1–7

[2] H.-J. Bandelt, A.W.M. Dress, A Canonical Decomposition Theory for Metrics on a Finite Set, *Advances in Mathematics* **92** (1992) 47–105

[3] P. Buneman, A Note on the Metric Properties of Trees, *Journal of Combinatorial Theory* (B) **17** (1974) 48–50

[4] X. Chen, V. Chvátal, Problems related to a de Bruijn-Erdős theorem, *Discrete Appl. Math.* **156** (2008) 2101–2108

[5] A.W.M. Dress, The category of X-nets, in: J. Feng, J. Jost, M. Quian (Eds.), Networks: From Biology to Theory, Springer, Berlin, 2007, pp. 271–289

[6] A.W.M. Dress, M. Krüger, Parsimonious phylogenetic trees in metric spaces and simulated annealing, *Advances in Applied Mathematics* **8** (1987) 8–37

[7] E. Howorka, On Metric Properties of Certain Clique Graphs, *Journal of Combinatorial Theory* (B) **27** (1979) 67–74

[8] D.C. Kay, G. Chartrand, A characterization of certain ptolemaic graphs, *Canad. J. Math.* **17** (1965) 342–346

[9] V. Mascioni, Equilateral Triangles in Finite Metric Spaces, *The Electric Journal of Combinatorics* **11** (2004) #R18

Measuring Graph Robustness via Game Theory

Dávid Szeszlér[1]

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
Magyar Tudósok krt. 2., Budapest, 1117,
Hungary
szeszler@cs.bme.hu

Abstract: Measuring the reliability of a network is one of the rich and complex areas of combinatorial optimization. Since the precise meaning of reliability highly depends on the application, there is an abundance of reliability metrics that have been proposed. Applying game-theoretical tools for measuring security has become very common. The basic idea is very natural: define a game between two virtual players, the Attacker and the Defender, such that the rules of the game capture the circumstances under which reliability is to be measured. Then analyzing the game might give rise to an appropriate security metric: the better the Attacker can do in the game, the lower the level of security is. This kind of analysis can give rise to new graph reliability metrics and in some cases it can shed a new light on some well-known ones. In this paper we survey a few recent results of this type.

Keywords: Robustness, Game Theory, Nash-equilibrium, Connectivity, Graph Strength

1 Introduction

The problem of measuring the robustness or reliability of a graph arises in many applications. The most widely applied reliability metrics are obviously the connectivity based ones, however, these are unsuitable in many cases. The reason for that is that in many applications the network is almost completely functional if removing some nodes or links results in the loss of only a small number of nodes that are in some sense insignificant or peripheral. Connectivity based metrics (even weighted versions of these) are not capable of capturing this idea as they are only concerned with whether the resulting graph is connected or not.

There is an abundance of recent books and papers on game-theoretical tools for measuring and increasing security. Since all aspects of security are obviously of utmost importance nowadays and game theory as a tool to address related problems presents itself very naturally, the literature on this topic is extremely diverse. Much of the arsenal of game theory has been employed on various applications which very often have little in common besides somehow being related to security. In this paper, however, only the theory of two-player, zero-sum games, the simplest and probably most widely known subfield of game theory will be relied on to address various problems raised by applications concerning the measuring of graph robustness.

All results mentioned in this paper will be based on the following approach. Assume that an input graph G is given (in some cases with a few designated vertices). G will either be directed or undirected depending on the application. Besides that, in most cases certain weight functions will also be part of

[1]Research is supported by the grant No. OTKA 108947 of the Hungarian National Research, Development and Innovation Office (NKFIH).

the input: for each edge $e \in E(G)$ or vertex $v \in V(G)$ the "damage" caused by the loss of e or v (or in other words, the "importance" of e or v) will be denoted by $d(e)$ or $d(v)$, respectively; furthermore, the cost of attacking an edge e will be denoted by $c(e)$. In each application, we will define a two-player, zero-sum game on G between two virtual players, the Attacker and the Defender. In all such games, the Attacker will choose (or "attack and destroy") an edge e of G (or more formally: the set of pure strategies of the Attacker will be $E(G)$). Simultaneously (or simply without knowing the Attacker's chosen edge) the Defender will choose a subset of the edges $Z \subseteq E(G)$ that will be thought of as some kind of "communication infrastructure" and the requirements on which will vary in each application (for example, Z can be the edge set of a path or a spanning tree, etc). Regardless of the Defender's choice, the Attacker will have to pay the cost of attack $c(e)$ to the Defender. There will be no further payoff if $e \notin Z$. If, on the other hand, $e \in Z$ then the Defender will pay the Attacker an amount that will be individually defined for each application (and will somehow depend on e, Z and the damage values $d(e)$ and $d(v)$). Since these games will be two-player, zero-sum games by definition, they will have a unique Nash-equilibrium payoff V (which will simply be referred to as the *game value* in this paper) by Neumann's classic Minimax Theorem (see Section 2). Since V is the highest expected gain the Attacker can guarantee himself by an appropriately chosen mixed strategy, it makes sense to say that the reciprocal of V is a valid reliability metric in the sense corresponding to the specific definition of the game.

We remark that it might seem unrealistic in the above described framework that the Defender should receive the cost of attack $c(e)$ from the Attacker (as the Defender is indifferent to the costs and efforts associated with an attack, she is only affected by the damage caused). In other words, it would be more natural to assume that the above given payoffs only describe the Attacker's gain while the Defender's loss depends exclusively on e, Z and the damage values $d(e)$ and $d(v)$ (and is thus always bigger by $c(e)$ than the Attacker's gain). This would also imply that the game is not zero-sum any more. However, it is easily shown that the thus-obtained non-zero-sum game is essentially equivalent to the zero-sum game described above. This equivalency is due to the fact that the sum of the payoffs only depends on the choice of the Attacker and it more precisely means that Nash-equilibria of the two versions of the game are identical and the Attacker's Nash-equilibrium payoff is unique in the non-zero-sum version of the game and it is equal to the (unique) Nash-equilibrium payoff corresponding to the zero-sum version. (An analogous statement would not be true for the Defender.) The proof of this equivalency is a simple exercise (see [10, Lemma 1] for a proof). We will disregard this point in the remainder of the paper and focus on the zero-sum game versions described above.

2 Preliminaries on Game Theory

In this section we very briefly summarize all the necessary background on game theory. A *(finite) two-player, zero-sum game* is given by a matrix M called the *payoff matrix*. Columns of M correspond to one of the players and rows of M to the other, so for the sake of simplicity one can refer to the two players as *Column Player* and *Row Player*. Columns and rows of M are called the *pure strategies* of the respective players. The matrix M defines the game in the following sense: both players choose one of their pure strategies (simultaneously, without knowing each other's choices) and then the corresponding entry of M (that is, the one in the intersection of the chosen row and column) is payed by the Row Player to the Column Player. (Obviously, a negative payment means that in reality it is the Column Player who pays the absolute value of the amount to the Row Player.)

A *mixed strategy* of a player is a probability distribution on their pure strategies. If M is a $k \times n$ matrix then it is natural to store the Column Player's and the Row Player's mixed strategies as n-dimensional column vectors and k-dimensional row vectors, respectively. If we fix a pair of mixed strategies $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^k$ then the Column Player's expected gain (or, equivalently, the Row Player's expected loss) is obviously $\mathbf{y}M\mathbf{x}$. It is sensible for the Column Player to choose a mixed strategy $\mathbf{x}$ that maximizes his worst case expected gain, therefore he is interested in finding an $\mathbf{x}$ that maximizes the minimum value of $\mathbf{y}M\mathbf{x}$ over all possible mixed strategies $\mathbf{y}$ of the Row Player; in other words, his job is $\max_{\mathbf{x}} \left\{ \min_{\mathbf{y}} \{\mathbf{y}M\mathbf{x}\} \right\}$.

Analogously, the Row Player's task is $\min_{\mathbf{y}} \left\{ \max_{\mathbf{x}} \{\mathbf{y}M\mathbf{x}\} \right\}$; that is, she wants to minimize her worst case expected loss. Neumann's classic Minimax Theorem [12] states that these two values are equal for every payoff matrix M: $\max_{\mathbf{x}} \left\{ \min_{\mathbf{y}} \{\mathbf{y}M\mathbf{x}\} \right\} = \min_{\mathbf{y}} \left\{ \max_{\mathbf{x}} \{\mathbf{y}M\mathbf{x}\} \right\}$. This common value is called the *game value* corresponding to M. Since a pair of mixed strategies $(\mathbf{x}, \mathbf{y})$ that attain the corresponding optima is equivalent to the (more general) notion of a Nash-equilibrium in the special case of two-player, zero-sum games, the game value is also referred to as a (Nash-)equilibrium payoff in the literature (which is known to be unique in this special case). However, in this paper we will keep calling it the game value.

It is useful to mention that the description of the tasks of the two players can be simplified by observing that it is sufficient for a mixed strategy to "guard against" all pure strategies of the other player, that will imply that it also guards against all mixed strategies. For example, if every entry of the column vector $M\mathbf{x}$ is at least μ for a mixed strategy $\mathbf{x}$, that translates to saying that no matter which pure strategy the Row Player picks, the Column Player's expected gain is at least μ. However, this also implies $\mathbf{y}M\mathbf{x} \geq \mu$ for every mixed strategy $\mathbf{y}$ (since $\mathbf{y}M\mathbf{x}$ is a convex combination of the entries of $M\mathbf{x}$). Hence the Column Player's task can also be described as maximizing the minimum entry of $M\mathbf{x}$ over all mixed strategies $\mathbf{x}$ (and the Row Player's case is analogous).

The above also implies (as it is shown in many textbooks, see e.g. [11]) that two-player, zero-sum games are easy to handle algorithmically via linear programming: optimum mixed strategies for the game given by M can be found efficiently by solving the following linear program and its dual:

$$\max\{\mu : M\mathbf{x} \geq \mu \cdot \mathbf{1}, \mathbf{1} \cdot \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}$$

(where $\mathbf{1}$ denotes the all-1 vector). However, since the size of the payoff matrix M will be exponential in the size of the input graph G in all applications mentioned in this paper, this approach will not be viable.

3 Connectivity Based Metrics

The following simple example might illuminate the approach described in the Introduction.

THE *st*-PATH GAME

Input: A connected, undirected graph G and two vertices $s, t \in V(G)$;
The Attacker chooses an edge e of G;
The Defender chooses a path P between s and t;
The Payoff from the Defender to the Attacker is 1 if e is on P and 0 otherwise.

The origin of the following simple claim is unclear, one can regard it as folklore.

Claim 1 *The game value of the st-path game is $\frac{1}{\lambda(s,t)}$, where $\lambda(s,t)$ denotes the edge-connectivity between s and t (that is, the size of the minimum cut separating s and t).*

PROOF: Let C be a cut of size $\lambda(s,t)$ that separates s and t and assume that the Attacker uses the following mixed strategy: he assigns a probability of $\frac{1}{\lambda(s,t)}$ to every edge of C and 0 to the rest of the edges. Since every st-path contains at least one edge from C, this mixed strategy guarantees the Attacker an expected gain of at least $\frac{1}{\lambda(s,t)}$. This proves that game value is at least $\frac{1}{\lambda(s,t)}$.

Now choose $\lambda(s,t)$ pairwise edge-disjoint paths between s and t (which are known to exist by Menger's classic theorem, see [13, Section 9.1]). Assume that the Defender uses the following mixed strategy: she assigns a probability of $\frac{1}{\lambda(s,t)}$ to every chosen path and 0 to the rest of the st-paths. Since the chosen paths are edge-disjoint, this mixed strategy guarantees the Defender an expected loss of at most $\frac{1}{\lambda(s,t)}$. Hence the game value is at most $\frac{1}{\lambda(s,t)}$. □

The relevance of Claim 1 is that it shows that the notion of edge-connectivity between two vertices (viewed as a reliability metric) is well captured by the st-path game. However, the notion of (general)

edge-connectivity (the minimum size $\lambda(G)$ of a subset of edges the removal of which disconnects G) is also easy to capture by a similar game:

THE PATH GAME

Input: A connected, undirected graph G;
First *the Attacker* chooses two distinct nodes $s, t \in V(G)$ and declares them to the Defender;
then *the Defender* chooses a path P between s and t,
and (simultaneously) *the Attacker* chooses an edge e of G;
The Payoff from the Defender to the Attacker is 1 if e is on P and 0 otherwise.

Claim 2 *The game value of the path game is* $\frac{1}{\lambda(G)}$.

PROOF: The proof is analogous to that of Claim 1 with the only difference being that the Attacker first chooses the nodes s and t in such a way that they are separated by a minimum cut of G. □

Obviously, the notion of node connectivity of G (either between two specific vertices or in general) can be captured by analogously defined games as the ones in Claims 1 and 2. The following theorem shows that, as one would expect, the weighted versions of these games lead to weighted minimum cuts.

THE WEIGHTED *st*-PATH GAME

Input: A connected, undirected graph G, two nodes $s, t \in V(G)$, a damage function $d : E(G) \to \mathbb{R}^+$ and a cost function $c : E(G) \to \mathbb{R}$;
The Attacker chooses an edge e of G;
The Defender chooses a path P between s and t;
The Payoff from the Defender to the Attacker is $d(e) - c(e)$ if e is on P and $-c(e)$ otherwise.

Obviously, the above payoffs correspond to the framework described in the Introduction: the cost of attack $c(e)$ must be paid by the Attacker in all cases, but he receives the damage value $d(e)$ if he succeeds in hitting the st-path chosen by the Defender.

The weighted st-path game is considered and solved in the $d(e) \equiv 1$ and the $c(e) \equiv 0$ cases in [6] and [15], respectively. (In [6], a generalization of the $d(e) \equiv 1$ case of the game is also solved: there the Attacker can target a subset of the edges of a given size and the Defender can choose two paths between two source-destination pairs.) The following result, however, seems to be new.

Theorem 3 *For every input of the weighted st-path game the game value is*

$$\max\left\{\left\{\frac{1-q(C)}{p(C)} : C \text{ is a cut that separates } s \text{ and } t\right\} \cup \left\{-c(e) : e \in E(G)\right\}\right\},$$

where $p(e) = \frac{1}{d(e)}$ *and* $q(e) = \frac{c(e)}{d(e)}$ *for all* $e \in E(G)$.

PROOF: Let the value of the above maximum be μ. Assume first that $\mu = -c(e)$ for some $e \in E(G)$. Then if the Attacker targets e with a probability of 1, his total expected gain is obviously at least μ. Now assume that $\mu = \frac{1-q(C)}{p(C)}$ for a cut C that separates s from t and let the Attacker use the following mixed strategy: assign a probability of $\frac{p(e)}{p(C)}$ to every edge of C and 0 to the rest of the edges. Consider an arbitrary path P between s and t and fix an edge $e \in C$. Then e contributes to the Attacker's expected gain by $\frac{p(e)}{p(C)}(d(e) - c(e)) = \frac{1-q(e)}{p(C)}$ or $\frac{p(e)}{p(C)}(-c(e)) = \frac{-q(e)}{p(C)}$ depending on whether e is on P or not, respectively. Since C obviously contains at least one edge of P, the Attacker's total expected gain is at least $\frac{1-q(C)}{p(C)} = \mu$. Since in all cases the Attacker has a mixed strategy that guarantees him an expected gain of at least μ, the game value is also at least μ.

Replace every edge $e = \{u, v\}$ of G by the directed arcs $e' = \overrightarrow{uv}$ and $e'' = \overrightarrow{vu}$ and let the capacity of both be $\mu \cdot p(e) + q(e)$. Then, by the definition of μ, the capacity of every edge is non-negative and the total

capacity of every $s\bar{t}$-cut is at least 1. Therefore there exists a flow f from s to t of overall value 1 by the Ford-Fulkerson theorem. Assume without loss of generality that for every edge e of G either $f(e') = 0$ or $f(e'') = 0$. It is well-known (see [13, Section 10.3]) that f is a non-negative linear combination of characteristic vectors of directed paths from s to t and directed cycles. This implies that there exists a set of (undirected) paths $P_1, P_2, \ldots, P_t$ in G between s and t and corresponding non-negative coefficients $\alpha_1, \alpha_2, \ldots, \alpha_t$ such that $\sum_{i=1}^{t} \alpha_i = 1$ and

$$\sum \{\alpha_i : e \text{ is on } P_i\} \leq \mu p(e) + q(e) \tag{1}$$

holds for each edge e.

Now assume that the Defender uses the following mixed strategy: for every $1 \leq i \leq t$ she assigns the probability α_i to P_i (and 0 to the rest of the st-paths). Then if the Attacker targets an edge e then her expected loss is

$$\sum_{i:e\in E(P_i)} \alpha_i(d(e) - c(e)) - \sum_{i:e\notin E(P_i)} \alpha_i c(e) = d(e)\left(\sum_{i:e\in E(P_i)} \alpha_i\right) - c(e) \leq d(e)(\mu p(e) + q(e)) - c(e) = \mu$$

by (1). Therefore the game value is at most μ. □

Corollary 4 ([15]) *If $c(e) = 0$ is assumed for all $e \in E(G)$ then the game value of the weighted st-path game is $\frac{1}{\lambda_p(s,t)}$, where $p(e) = \frac{1}{d(e)}$ for every edge e and $\lambda_p(s,t)$ is the minimum total weight of a cut that separates s and t with respect to the weight function p.*

4 Graph Strength and Related Metrics

The *strength* of a connected graph G was defined by Gusfield [9]. The idea is quite natural: if we remove a subset $U \subseteq E(G)$ of the edges then the efficiency of this "attack" against G can be measured by the ratio of the number of new components created and $|U|$ (that is, the "effort" required for the attack). Then it makes sense to define the reciprocal of the maximum efficiency of an attack to be a security metric: $\sigma(G) = \min\left\{\frac{|U|}{\text{comp}(G-U)-1} : U \subseteq E(G), \text{comp}(G-U) > 1\right\}$, where $\text{comp}(G-U)$ is the number of components of the graph obtained from G by deleting U. This notion was extended to a weighted version by Cunningham [7]:

Definition 5 *Assume that a connected graph G is given with a positive weight function $p : E(G) \to \mathbb{R}^+$ on its edges. Then*

$$\sigma_p(G) = \min\left\{\frac{p(U)}{\text{comp}(G-U)-1} : U \subseteq E(G), \text{comp}(G-U) > 1\right\}$$

is called the strength *of G with respect to p.*

$\sigma_p(G)$ is computable in strongly polynomial time as it was shown by Cunningham [7]. It was proved in [14] that the following game is capable of capturing the notion of $\sigma_p(G)$. The game resembles the weighted st-path game defined above with the only difference being that the Defender's pure strategies are spanning trees instead of st-paths.

The spanning tree game

Input: A connected, undirected graph G, a damage function $d : E(G) \to \mathbb{R}^+$ and a cost function $c : E(G) \to \mathbb{R}$;

The Attacker chooses an edge e of G;

The Defender chooses a spanning tree T of G;

The Payoff from the Defender to the Attacker is $d(e) - c(e)$ if e is in T and $-c(e)$ otherwise.

Theorem 6 ([14]) *For every input of the spanning tree game the game value is*

$$\max_{\emptyset \neq U \subseteq E(G)} \frac{\mathrm{comp}(G-U)-1-q(U)}{p(U)},$$

where $p(e)=\frac{1}{d(e)}$ *and* $q(e)=\frac{c(e)}{d(e)}$ *for all* $e \in E(G)$ *(and* $\mathrm{comp}(G-U)$ *is the number of components of the graph obtained from* G *by deleting* U*). Furthermore, there exists a strongly polynomial algorithm that computes the game value of the spanning tree game and an optimum mixed strategy for both players.*

We remark that the above formula (without a corresponding strongly polynomial algorithm) was shown previously in the special case of $d(e) \equiv 1$ in [8].

Corollary 7 ([14]) *The game value of the spanning tree game is* $\frac{1}{\sigma_p(G)}$ *if* $p(e)=\frac{1}{d(e)}$ *and* $c(e)=0$ *is assumed for all* $e \in E(G)$*.*

It is important to mention that Theorem 6 was proved in [14] in a much more general, matroidal setting: the MATROID BASE GAME was defined analogously to the spanning tree game with the only difference being that the Attacker chooses an element of the ground set S of a given matroid $M=(S,\mathcal{I})$ and the Defender chooses a base B of M. Then the following was proved:

Theorem 8 ([14]) *Assume that the matroid* $M=(S,\mathcal{I})$ *and damage and cost functions* $d: S \to \mathbb{R}^+$ *and* $c: S \to \mathbb{R}$*, respectively are given. Then the game value of the matroid base game is equal to*

$$\max_{\emptyset \neq U \subseteq S} \frac{\mathrm{r}(S)-\mathrm{r}(S-U)-q(U)}{p(U)},$$

where $p(s)=\frac{1}{d(s)}$ *and* $q(s)=\frac{c(s)}{d(s)}$ *for all* $s \in S$*. Furthermore, if* M *is given by an independence testing oracle then there exists a strongly polynomial algorithm that computes the game value of the matroid base game and an optimum mixed strategy for both players.*

We remark that the above theorem was essentially known before in the special case of $c \equiv 0$: then one easily shows that solving the matroid base game is equivalent to the *capacitated fractional base packing* problem discussed in [13, Section 42.4], where a strongly polynomial algorithm is given in [13, Theorem 42.7]. However, that algorithm does not seem to generalize to the $c \not\equiv 0$ case. Hence the above theorem can also be regarded as a generalization of [13, Theorem 42.7].

Obviously, Theorem 8 gives rise to a number of natural extensions of the spanning tree game and readily provides the corresponding modifications of the notion of graph strength. For example, one could modify the definition of the spanning tree game by allowing the Defender to choose the edge set of the union of k edge-disjoint spanning trees (where $k \geq 1$ is given); or the Defender could choose a spanning edge set of a given size, etc. (Besides that, the matroid base game turned out to be relevant even in the very special case of the uniform matroid: that came up in an application concerning the security of content-adaptive steganography, see [10].)

In [14] a further generalization of the matroid base game was also considered: the COMMON BASE GAME is almost identical to the matroid base game with the only difference being that the Defender chooses a common base of two matroids given on the common ground set S. Not all results on the matroid base game seem to extend smoothly to the common base game, a strongly polynomial algorithm is only known is certain special cases. We omit the technical details here (see [14]), we only discuss an application of the common base game that yields a new security metric, a directed analogue of graph strength.

Assume that a digraph G is given. Call a subset of the nodes $R \subseteq V(G)$ a *source set* if every node of G is reachable from a node in R via a directed path. A vertex $r \in V(G)$ is a *source node* if $\{r\}$ is a single-element source set. For every arc set $U \subseteq E(G)$, denote by $\mathrm{source}(G-U)$ the minimum cardinality of a source set in the digraph obtained from G by deleting U. (In other words, $\mathrm{source}(G-U)$ is the number of weak components in a maximum size branching of $G-U$.)

Definition 9 *Assume that a directed graph G is given that has a source node; assume further that a positive weight function $p : E(G) \to \mathbb{R}^+$ is given. Then*

$$\overrightarrow{\sigma}_p(G) = \min\left\{\frac{p(U)}{\text{source}(G-U)-1} : U \subseteq S, \text{source}(G-U) > 1\right\}$$

is the directed strength *of G with respect to p.*

It was proved in [14] that $\overrightarrow{\sigma}_p(G)$ is computable is strongly polynomial time.

Recall that an *arborescence* of G is a subset A of the arcs that is a spanning tree of the underlying undirected graph such that the digraph $(V(G), A)$ has a source node. (It is well-known and elementary that the existence of an arborescence is equivalent to the existence of a source node.) Then the following is a straightforward analogue of the spanning tree game:

The arborescence game

Input: A directed graph G that has a source node, a damage function $d : E(G) \to \mathbb{R}^+$ and a cost function $c : E(G) \to \mathbb{R}$;

The Attacker chooses an arc e of G;

The Defender chooses an arborescence A of G;

The Payoff from the Defender to the Attacker is $d(e) - c(e)$ if e is in A and $-c(e)$ otherwise.

The following theorem yields an analogous description of the game value as that of Theorem 6.

Theorem 10 ([14]) *For every input of the arborescence game the game value is*

$$\max_{\emptyset \neq U \subseteq E(G)} \frac{\text{source}(G-U) - 1 - q(U)}{p(U)},$$

where $p(e) = \frac{1}{d(e)}$ and $q(e) = \frac{c(e)}{d(e)}$ for all $e \in E(G)$.

As a corollary, we get the analogue of the connection between graph strength and the spanning tree game.

Corollary 11 ([14]) *The game value of the arborescence game is $\frac{1}{\overrightarrow{\sigma}_p(G)}$ if $p(e) = \frac{1}{d(e)}$ and $c(e) = 0$ is assumed for all $e \in E(G)$.*

5 Persistence and Related Metrics

In [7] another reliability metric was defined that is based on a somewhat similar idea to that of graph strength. Assume that a "headquarters" node $r \in V(G)$ is given in a graph G with the special role that every node needs to communicate with r. Assume further that the "importance" of the information stored in a node v is represented by $d(v)$. Then if a subset U of the edges is removed, the efficiency of this "attack" can be measured by the ratio of $|U|$ and the total d-value of the nodes that become unreachable from r. Hence the reciprocal of the maximum efficiency of an attack is again a sensible a reliability metric:

Definition 12 *Assume that a connected graph G, a designated node $r \in V(G)$ and a non-negative weight function $d : (V(G) \setminus \{r\}) \to \mathbb{R}^{\geq 0}$ are given. Then*

$$\pi(G) = \min\left\{\frac{|U|}{\lambda(U)} : U \subseteq E(G), \lambda(U) > 0\right\}$$

is called the persistence *of G, where*

$$\lambda(U) = \sum\big\{d(v) : v \text{ is unreachable from } r \text{ after removing } U\big\}.$$

It is important to remark the following to avoid confusion. In [7] the above notion was defined on directed graphs (since, although both versions make sense, the directed one generalizes the undirected version by the usual trick of replacing every undirected edge by two directed ones). The notion was not given any specific name, it was regarded as a directed analogue of graph strength $\sigma_p(G)$ (see Definition 5) and was referred to as the "directed model". However, the above notion of $\pi(G)$ is substantially different from and not to be confused with *directed strength* $\overrightarrow{\sigma}_p(G)$ defined in Definition 9. In fact, we believe that $\overrightarrow{\sigma}_p(G)$ is a much closer analogue to $\sigma_p(G)$ than $\pi(G)$. The name *persistence* was coined in [1] for easier reference (and to better distinguish $\pi(G)$ from $\sigma_p(G)$ and $\overrightarrow{\sigma}_p(G)$).

It was proved in [7] that $\pi(G)$ is computable in strongly polynomial time (see [5] for a somewhat simpler description of the algorithm). Despite its apparent naturality and simplicity, the notion of $\pi(G)$ did not seem to receive much attention for more than two decades when it came up in an application: it was argued in [1] and [4] that it is very appropriate for measuring the reliability of wireless sensor networks.

The following game is related to the spanning tree game, but it is capable of capturing the notion of persistence (in the sense seen above). The idea is again natural: if the Attacker targets the edge e and succeeds in hitting the spanning tree T chosen by the Defender, then his gain is assumed to be the total damage he causes: the sum of the d-values of all nodes in the component of $T - e$ not containing r.

THE ROOTED SPANNING TREE GAME

Input: A connected, undirected graph G, a cost function $c : E(G) \to \mathbb{R}$, a node $r \in V(G)$ and a damage function $d : (V(G) \setminus \{r\}) \to \mathbb{R}^{\geq 0}$;

The Attacker chooses an edge e of G;

The Defender chooses a spanning tree T of G;

The Payoff from the Defender to the Attacker is $\lambda(T, e) - c(e)$, if e is in T and $-c(e)$ otherwise, where

$$\lambda(T, e) = \sum \big\{ d(v) : v \text{ is unreachable from } r \text{ in } T \text{ after removing } e \big\}.$$

Theorem 13 ([2]) *For every input of the rooted spanning tree game, the game value is*

$$\max_{\emptyset \neq U \subseteq E(G)} \frac{\lambda(U) - c(U)}{|U|}.$$

Corollary 14 ([2]) *The game value of the rooted spanning tree game is* $\frac{1}{\pi(G)}$ *if* $c(e) = 0$ *is assumed for all* $e \in E(G)$.

In fact, it is only Corollary 14 that is explicitly stated in [2], however, the proof extends smoothly to show Theorem 13 (we omit the details here). Furthermore, there are two further possible extensions of the rooted spanning tree game and Theorem 13, both follow from trivial modifications of the proof in [2]. The first one is to consider directed graphs instead of undirected ones: the notion of persistence is straightforward in this case (and, as already mentioned above, generalizes the undirected case) and the only change needed in the corresponding game is to consider arborescences rooted in r instead of spanning trees. Secondly, in [7] the notion of $\pi(G)$ was originally defined in a weighted version: $\pi_s(G) = \min \left\{ \frac{s(U)}{\lambda(U)} : U \subseteq E(G), \lambda(U) > 0 \right\}$, where $s : E(G) \to \mathbb{R}^+$ is a given positive weight function. In order to capture this weighted version of persistence, the definition of the rooted spanning tree game would be needed to be modified in the following way: the Attacker's gain would be $\frac{\lambda(T,e)}{s(e)} - c(e)$ if he hits the Defender's spanning tree T (and $-c(e)$ otherwise). Admittedly, there does not seem to be any natural intuition behind this definition, however, the game value in this case turns out to be $\max_{\emptyset \neq U \subseteq E(G)} \frac{\lambda(U) - q(U)}{s(U)}$, where $q(e) = s(e) \cdot c(e)$ for all e, which does indeed yield the reciprocal of $\pi_s(G)$ in the $c(e) \equiv 0$ case.

To conclude this paper, we mention an interesting variant of the rooted spanning tree game (only in its simplest case, with no weight functions) that lacks the role of a special headquarters node. The intuitive idea behind the definition is quite natural: if the Attacker succeeds in hitting the spanning tree chosen by the Defender then the bigger component of $T-e$ can still be regarded as a functioning network, so the Defender's loss (and thus the Attacker's gain) is the size of the smaller component.

The "you decide, I choose" spanning tree game

Input: A connected, undirected graph G;
The Attacker chooses an edge e of G;
The Defender chooses a spanning tree T of G;
The Payoff from the Defender to the Attacker is the number of nodes in the *smaller* component of $T-e$ if e is in T and 0 otherwise.

The following theorem expresses the game value of the above game in terms of a special multicommodity flow problem.

Theorem 15 ([3]) *Consider the following multicommodity flow problem for an arbitrary undirected, connected graph G. A flow f_v corresponds to every vertex $v \in V(G)$; the target node for f_v is v and every other vertex is assumed to be a source node for f_v which all produce a common flow amount of α_v. All edges of G have a capacity of 1 and they are all undirected. (In other words: every edge can carry commodities in both directions, but the total value of all carried amounts in both directions add up to at most 1.) The objective is to maximize $\sum_{v \in V(G)} \alpha_v$. Then if we denote the maximum value of this problem by $\mu(G)$ then the game value of the "you decide, I choose" spanning tree game is $\frac{1}{\mu(G)}$.*

The above theorem suggests that $\frac{1}{\mu(G)}$ could be regarded as a special, new graph reliability metric (that corresponds to the intuition behind the "you decide, I choose" spanning tree game). Obviously, $\mu(G)$ can be computed in polynomial time via linear programming. However, not much more than that is known about $\mu(G)$ (see [3] for some further details).

References

[1] A. Laszka, L. Buttyán and D. Szeszlér, Optimal Selection of Sink Nodes in Wireless Sensor Networks in Adversarial Environments, *Proc. of 2nd IEEE International Workshop on Data Security and Privacy in wireless Networks (D-SPAN), Lucca, Italy*, pp. 1-6 (2011).

[2] A. Laszka, D. Szeszlér and L. Buttyán, Game-theoretic Robustness of Many-to-one Networks, *Proc. of Game Theory for Networks: Third International ICST Conference, GameNets 2012, Vancouver, Canada*, pp. 88-98, Springer Berlin Heidelberg (2012).

[3] A. Laszka, D. Szeszlér and L. Buttyán, Linear Loss Function for the Network Blocking Game: An Efficient Model for Measuring Network Robustness and Link Criticality *Proc. of Decision and Game Theory for Security: Third International Conference, GameSec 2012, Budapest, Hungary*, pp. 152-170, Springer Berlin Heidelberg (2012).

[4] A. Laszka, L. Buttyán and D. Szeszlér, Designing robust network topologies for wireless sensor networks in adversarial environments, *Pervasive and Mobile Computing*, **9(4)**, pp. 546-563, (2013).

[5] L. Buttyán, A. Laszka and D. Szeszlér, A Minimum Cost Source Location Problem for Wireless Sensor Networks, *Proc. 8th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications, Veszprm, Hungary*, pp. 79-88 (2013).

[6] G. Calinescu, S. Kapoor, M. Quinn and J. Shin, Adversary Games in Secure/Reliable Network Routing, *Proc. of Game Theory for Networks: Second International ICST Conference, GameNets 2011, Shanghai, China*, pp. 249-264, Springer Berlin Heidelberg (2012).

[7] W. H. Cunningham, Optimal attack and reinforcement of a network, *Journal of the ACM (JACM)* **32(3)**, pp. 549-561 (1985).

[8] A. Gueye, J. C. Walrand and V. Anantharam, A network topology design game: How to choose communication links in an adversarial environment, *Proc. of the 2nd International ICST Conference on Game Theory for Networks, GameNets* **11**, (2011).

[9] D. Gusfield, Connectivity and edge-disjoint spanning trees, *Information Processing Letters* **16.2**, pp. 87-89 (1983).

[10] A. Laszka and D. Szeszlér, Hide and Seek in Digital Communication: The Steganography Game, *Proc. 9th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, Fukuoka, Japan*, pp. 126-136 (2015).

[11] J. Matoušek and B. Gärtner, Understanding and Using Linear Programming, Springer, Berlin, Heidelberg (2007).

[12] J. v. Neumann, Zur Theorie der Gesellschaftsspiele, *Mathematische Annalen* **100(1)**, pp. 295-320 (1928).

[13] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, Algorithms and Combinatorics **24**, Springer, Berlin, Heidelberg (2003).

[14] D. Szeszlér, Security games on matroids, *Mathematical Programming* **161(1)**, pp. 347-364, (2017).

[15] A. Washburn and K. Wood, Two-Person Zero-Sum Games for Network Interdiction, *Operations Research* **43(2)**, pp. 243-251, (1995).

Excluded t-factors in Bipartite Graphs: A Unified Framework for Nonbipartite Matchings and Restricted 2-matchings[1]

Kenjiro Takazawa[2]

Department of Industrial and Systems Engineering
Faculty of Science and Engineering
Hosei University, Tokyo 184-8584, Japan
takazawa@hosei.ac.jp

Abstract: We propose a new framework of optimal t-matchings excluding prescribed t-factors in bipartite graphs. It is a generalization of the nonbipartite matching problem and includes a number of generalizations such as the triangle-free 2-matching, square-free 2-matching, and even factor problems. We demonstrate a unified understanding of those generalizations by designing a combinatorial algorithm for our problem under a reasonable assumption, which is broad enough to include the specific problems listed above. We first present a min-max theorem and a combinatorial algorithm for the unweighted version. We further provide a linear programming formulation with dual integrality and a primal-dual algorithm for the weighted version. A key ingredient of our algorithm is a technique of shrinking forbidden structures, which commonly extends the techniques of shrinking odd cycles, triangles, and squares in Edmonds' blossom algorithm, in the triangle-free 2-matching algorithm, and in the square-free 2-matching algorithm, respectively.

Keywords: Nonbipartite Matching, Triangle-free 2-matching, Square-free 2-matching, Min-max Theorem, Dual-integral LP formulation, Combinatorial Algorithm

1 Introduction

Since matching theory [17] was established, a number of generalizations of the matching problem have been proposed up to the present date. Examples include path-matchings [4], even factors [5, 20], triangle-free 2-matchings [3, 19], simple square-free 2-matchings [10, 20], simple $K_{t,t}$-free t-matchings [8], simple K_{t+1}-free t-matchings [1], 2-matchings covering prescribed edge cuts [2, 12], and $\mathcal{U}$-feasible 2-matchings [25]. For most of those generalizations, important results in matching theory, such as a min-max theorem, polynomial algorithms, and a linear programming formulation with dual integrality, are extended. However, while some similar structures are found, in most cases they are studied separately and little connection among them is discovered.

In the present paper, we propose a new framework of *optimal t-matchings excluding prescribed t-factors*, to demonstrate a unified understanding of those generalizations. Our framework includes all of the generalizations listed above, and the traveling salesman problem (TSP) as well. This implies some intractability of the framework, but we propose a tractable class which includes many of the efficiently solvable classes of the above problems. Our main contribution is a min-max theorem and a combinatorial polynomial algorithm which commonly extend those for the matching and triangle-free 2-matching problems in nonbipartite graphs and the simple square-free 2-matching and $K_{t,t}$-free t-matching problems in bipartite graphs.

[1]The original version of this extended abstract appears in Proceedings of the 19th IPCO.
[2]Partially supported by JSPS/MEXT KAKENHI Grant Numbers 16K16012 and 25280004.

A key ingredient of our algorithm is a technique of shrinking excluded t-factors. This technique commonly extends the techniques of shrinking odd cycles, triangles, and squares in a matching algorithm [7], in a triangle-free 2-matching algorithm [3], and in square-free 2-matching algorithms in bipartite graphs [10, 20], respectively. We demonstrate that our framework is tractable in the class where this shrinking technique works.

1.1 Previous Work

The problems most relevant to our work are the *even factor*, *triangle-free* 2*-matching*, and *simple square-free* 2*-matching problems*.

The even factor problem [5] is a generalization of the nonbipartite matching problem, which admits a further generalization: the basic/independent even factor problem [5, 11] is its generalization including matroid intersection as well. Let $D = (V, A)$ be a digraph. A subset of arcs $F \subseteq A$ is a *path-cycle factor* if it is a vertex-disjoint collection of directed cycles (dicycles) and directed paths (dipaths). Equivalently, an arc subset F is a path-cycle factor if, in the subgraph (V, F), the indegree and outdegree of every vertex are at most one. An *even factor* is a path-cycle factor excluding dicycles of odd length (odd dicycles).

While the maximum even factor problem is NP-hard, in *odd-cycle symmetric* digraphs it enjoys min-max theorems [5, 21], an Edmonds-Gallai decomposition [21], and polynomial-time algorithms [5, 20]. A digraph is called *odd-cycle symmetric* if every odd dicycle has its reverse dicycle. Moreover, a maximum-weight even factor can be found in polynomial time in odd-cycle symmetric weighted digraphs, which are odd-cycle symmetric digraphs with arc-weight such that the total weight of the arcs in an odd dicycle is equal to that of its reverse dicycle [14, 22]. The maximum-weight matching problem is straightforwardly reduced to the maximum-weight even factor problem in odd-cycle symmetric weighted digraphs. The assumption of odd-cycle symmetry of (weighted) digraphs is justified by its relation to discrete convexity [16].

The triangle-free 2-matching and simple square-free 2-matching problems are examples of the *restricted* 2*-matching problem*, a main objective of which is to provide a tight relaxation of the TSP. Let $G = (V, E)$ be an undirected graph which may have parallel edges but may not have loops. For a positive integer t, an edge set $F \subseteq E$ is called a *t-matching* (resp., *t-factor*) if every vertex in V has at most (resp., exactly) t incident edges in F. A 2-matching is called *triangle-free* if it excludes cycles of length three. Note that a triangle-free 2-matching may contain parallel edges. For the maximum-weight triangle-free 2-matching problem in which every edge has its parallel copy with the same weight, a combinatorial algorithm together with a totally dual integral formulation is presented in [3, 19].

An edge set is called *simple* if it excludes parallel edges. If we restrict 2-matchings to be simple, the triangle-free 2-matching problem becomes much more complicated [9]. More generally, for a positive integer k, a simple 2-matching is called *$C_{\leq k}$-free* if it excludes cycles of length at most k. Finding a maximum simple $C_{\leq k}$-free 2-matching is NP-hard for $k \geq 5$, and is open for $k = 4$. In contrast, in bipartite graphs, the simple $C_{\leq 4}$-free 2-matching problem becomes tractable. We often refer to a simple $C_{\leq 4}$-free 2-matching in a bipartite graph as a *square-free* 2*-matching*. Throughout this paper, a square-free 2-matching always means a simple $C_{\leq 4}$-free 2-matching in a bipartite graph, unless otherwise stated.

For the square-free 2-matching problem, min-max theorems [8, 13], combinatorial algorithms [10, 20], and decomposition theorems [24] are established. For the weighted case, while finding a maximum-weight square-free 2-matching in a bipartite graph is NP-hard, it is solvable in polynomial time if the weight is *vertex-induced* on each C_4 [18, 23]. This assumption on the weight is again justified by its relation to discrete convexity [15].

It should be noted that Pap [20] presented combinatorial algorithms for the even factor and square-free 2-matching problems in the same paper. Indeed, these algorithms are based on similar techniques of shrinking odd cycles and C_4's, and may imply some similarity of these two problems. However, to the best of our knowledge, a comprehensive theory including both of these problems has not been proposed.

1.2 Our Contribution

In the present paper, we discuss *$\mathcal{U}$-feasible t-matchings*: for an undirected graph $G = (V, E)$ and $\mathcal{U} \subseteq 2^V$, a t-matching F is *$\mathcal{U}$-feasible* if it excludes a t-factor in U for each $U \in \mathcal{U}$ (see Definition 1 for a formal description). The optimal $\mathcal{U}$-feasible t-matching problem generalizes not only the $\mathcal{U}$-feasible 2-matching problem [25], but also all of the aforementioned generalizations of the matching problem. Thus, it could be recognized that $\mathcal{U}$-feasibility is a common generalization of the blossom constraint for the nonbipartite matching problem and the subtour elimination constraint for the TSP.

A main contribution of this paper is a min-max theorem and an efficient combinatorial algorithm for the maximum $\mathcal{U}$-feasible t-matching problem in bipartite graphs under a plausible assumption. Our algorithm runs in $O(t(|V|^4\alpha+|V|^3\beta+|V|^2|E|))$ time, where α and β are the time for expanding the shrunk structures and checking feasibility of an edge set, respectively. The complexities α and β are typically small, i.e., constant or $O(|V|)$, in the above specific cases. We further establish a linear programming description with dual integrality and a primal-dual algorithm for the maximum-weight $\mathcal{U}$-feasible t-matching problem in bipartite graphs, under the same plausible assumption. The complexity of the algorithm is $O(t(|V|^4\alpha + |V|^3(|E| + \beta)))$ time.

Imposing some assumption on $(G,\mathcal{U},t)$ would be reasonable in order to have $\mathcal{U}$-feasible t-matchings tractable. (Recall that it can describe Hamilton cycles.) Indeed, we assume for the excluded t-factors that the expanding technique is always valid (see Definition 4). This assumption is broad enough to include the instances reduced from nonbipartite matchings, even factors in odd-cycle symmetric digraphs, triangle-free 2-matchings in nonbipartite graphs, square-free 2-matchings, and simple $K_{t,t}$-free t-matchings in bipartite graphs. It would be noteworthy that the $\mathcal{U}$-feasible t-matching problem in *bipartite* graphs is a generalization of the *nonbipartite* matching problem.

For the weighted case, we also assume that the edge weights are *vertex-induced* for each $U \in \mathcal{U}$ (see Definition 2). We note that this assumption exactly corresponds to the previous assumptions for the maximum-weight even factor, square-free 2-matching, and simple $K_{t,t}$-free t-matching problems. Those previous assumptions are plausible from the viewpoint of discrete convexity [15, 16]. This would be an example of a unified understanding of the previous results on even factors and square-free 2-matchings.

2 Our Framework

Let $G = (V, E)$ be an undirected graph which may have parallel edges. An edge e connecting $u, v \in V$ is denoted by $\{u, v\}$. If G is a digraph, then an arc from u to v is denoted by (u, v). For $X \subseteq V$, let $G[X] = (X, E[X])$ denote the subgraph of G induced by X, that is, $E[X] = \{\{u, v\} \mid u, v \in X, \{u, v\} \in E\}$. Similarly, for $F \subseteq E$, define $F[X] = \{\{u, v\} \mid u, v \in X, \{u, v\} \in F\}$. If $X, Y \subseteq V$ are disjoint, then $F[X, Y]$ denotes the set of edges in F connecting X and Y.

For $v \in V$, let $\delta(v) \subseteq E$ denote the set of edges incident to v. For $F \subseteq E$ and $v \in V$, let $\deg_F(v) = |F \cap \delta(v)|$. Recall that F is a *t-matching* if $\deg_F(v) \leq t$ for each $v \in V$, and a *t-factor* if $\deg_F(v) = t$ for every $v \in V$.

Definition 1 *For a graph $G = (V, E)$ and $\mathcal{U} \subseteq 2^V$, a t-matching $F \subseteq E$ is called* $\mathcal{U}$-feasible *if $|F[U]| \leq \lfloor (t|U| - 1)/2 \rfloor$ for each $U \in \mathcal{U}$.*

Equivalently, a t-matching F in G is not $\mathcal{U}$-feasible if $F[U]$ is a t-factor in $G[U]$ for some $U \in \mathcal{U}$. This concept is a generalization of that for $\mathcal{U}$-feasible 2-matchings introduced in [25].

In what follows, we consider the maximum $\mathcal{U}$-feasible t-matching problem, in which the goal is to find a $\mathcal{U}$-feasible t-matching F maximizing $|F|$. We further deal with the maximum-weight $\mathcal{U}$-feasible t-matching problem, in which the objective is to find a $\mathcal{U}$-feasible t-matching F maximizing $w(F) = \sum_{e \in F} w(e)$ for a given edge-weight vector $w \in \mathbf{R}^E_+$. For a vector $x \in \mathbf{R}^E$ and $F \subseteq E$, in general we denote $x(F) = \sum_{e \in F} x(e)$. In discussing the weighted version, we assume that w is *vertex-induced on each $U \in \mathcal{U}$*.

Definition 2 *For a graph $G = (V, E)$, a vertex subset $U \subseteq V$, and an edge-weight $w \in \mathbf{R}^E$, w is called* vertex-induced on U *if there exists a function $\pi_U : U \to \mathbf{R}$ on U such that $w(\{u, v\}) = \pi_U(u) + \pi_U(v)$ for each $\{u, v\} \in E[U]$.*

3 Maximum $\mathcal{U}$-feasible t-matching

In this section, we exhibit a min-max theorem and a combinatorial algorithm scheme for the maximum $\mathcal{U}$-feasible t-matching problem in bipartite graphs. Our algorithm commonly extends those for nonbipartite matchings [7], even factors [20], triangle-free 2-matchings [3], and square-free 2-matchings [10, 20].

3.1 Weak Duality

Let $G = (V, E)$ be an undirected graph and $\mathcal{U} \subseteq 2^V$. For $X \subseteq V$, define $\mathcal{U}_X \subseteq \mathcal{U}$ and $C_X \subseteq X$ by $\mathcal{U}_X = \{U \in \mathcal{U} \mid U \text{ forms a component in } G[X]\}$, and $C_X = X \setminus \bigcup_{U \in \mathcal{U}_X} U$. Then the following inequality holds for an arbitrary $\mathcal{U}$-feasible t-matching and $X \subseteq V$. Note that the graph G do not need to be bipartite.

Lemma 3 *Let $G = (V, E)$ be an undirected graph, $\mathcal{U} \subseteq 2^V$, and t be a positive integer. For an arbitrary $\mathcal{U}$-feasible t-matching $F \subseteq E$ and $X \subseteq V$, it holds that*

$$|F| \leq t|X| + |E[C_{V \setminus X}]| + \sum_{U \in \mathcal{U}_{V \setminus X}} \left\lfloor \frac{t|U| - 1}{2} \right\rfloor. \tag{1}$$

Proof: The lemma follows from

$$2|F[X]| + |F[X, V \setminus X]| \leq t|X|, \tag{2}$$

$$|F[V \setminus X]| \leq |E[C_{V \setminus X}]| + \sum_{U \in \mathcal{U}_{V \setminus X}} \left\lfloor \frac{t|U| - 1}{2} \right\rfloor. \tag{3}$$

□

3.2 Algorithm

From now on, we assume bipartiteness of the graph. Let $G = (V, E)$ be an undirected bipartite graph. Denote the two color classes of V by V^+ and V^-. For $X \subseteq V$, denote $X^+ = X \cap V^+$ and $X^- = X \cap V^-$. The endvertices of an edge $e \in E$ in V^+ and V^- are denoted by $\partial^+ e$ and $\partial^- e$, respectively.

We begin with the description of shrinking a forbidden structure $U \in \mathcal{U}$. For concise notation, we denote the input graph by $\hat{G} = (\hat{V}, \hat{E})$ and the graph in hand, i.e., the graph resulted from possibly repeated shrinkings, by $G = (V, E)$. Consequently, we have that $\mathcal{U} \in 2^{\hat{V}}$. Denote the solution in hand by $F \subseteq E$. Intuitively, shrinking of U consists of identifying all vertices in U^+ and in U^- to obtain new vertices u_U^+ and v_U^-, respectively, and deleting all the edges in $E[U]$. A formal description is as follows.

Procedure Shrink(U)**.** Let u_U^+ and v_U^- be new vertices, and reset the endvertices of an edge $e \in \hat{E} \setminus \hat{E}[U]$ with $\partial^+ e = u$ and $\partial^- e = v$ as $\partial^+ e := u_U^+$ if $u \in U^+$ and $\partial^- e := v_U^-$ if $v \in U^-$. Then update G by $V^+ := (V^+ \setminus U^+) \cup \{u_U^+\}$, $V^- := (V^- \setminus U^-) \cup \{v_U^-\}$, and $E := E \setminus \hat{E}[U]$. Finally, $F := F \cap E$ and return (G, F).

We refer to a vertex $v \in V$ as a *natural vertex* if v is a vertex in the original graph $\hat{G}$, and as a *pseudovertex* if it is a newly added vertex in shrinking some $U \in \mathcal{U}$. We denote the set of the natural vertices by V_{n}, and that of the pseudovertices by V_{p}. For $X \subseteq \hat{V}$, define $X_{\mathrm{n}} = X \cap V_{\mathrm{n}}$ and

$X_{\rm p} = \bigcup\{u_U^+, v_U^- \mid U \subseteq X,\, u_U^+, v_U^- \in V_{\rm p}\}$. For $X \subseteq V$, define $\hat{X} \subseteq \hat{V}$ by $\hat{X} = X_{\rm n} \cup \bigcup\{U^+ \mid u_U^+ \in X \cap V_{\rm p}\} \cup \bigcup\{U^- \mid v_U^- \in X \cap V_{\rm p}\}$.

Procedure EXPAND(G, F) is to execute the reverse operation of SHRINK(U) for all shrunk $U \in \mathcal{U}$. A key point is that $\lfloor (t|U|-1)/2 \rfloor$ edges are added to F from $\hat{E}[U]$ for each $U \in \mathcal{U}$.

Procedure Expand(G, F)**.** Let $G := \hat{G}$. For each inclusionwise maximal $U \in \mathcal{U}$ which is shrunk, add $F_U \subseteq \hat{E}[U]$ of $\lfloor (t|U|-1)/2 \rfloor$ edges to F, so that F is a $\mathcal{U}$-feasible t-matching in $\hat{G}$. Now return (G, F).

The existence of F_U in Procedure EXPAND(G, F) is not trivial. In order to attain that $\hat{F} = F \cup \bigcup\{F_U \mid U \in \mathcal{U}$ is a maximal shrunk set$\}$ is a t-matching in $\hat{G}$, $F \subseteq E$ and $F_U \subseteq \hat{E}[U]$ should satisfy

$$\deg_F(u) \leq \begin{cases} t & (u \in V_{\rm n}), \\ 1 & (u \in V_{\rm p}) \end{cases} \tag{4}$$

$$\deg_{F_U}(u) \begin{cases} = t-1 & (u \text{ is incident to an edge in } F[U, V \setminus U]), \\ \leq t & (\text{otherwise}). \end{cases} \tag{5}$$

To achieve this, we maintain that F satisfies the degree constraint (4). Moreover, we assume that there exists F_U satisfying $|F_U| = \lfloor (t|U|-1)/2 \rfloor$ and (5) for an arbitrary F with (4) and every maximal shrunk set $U \in \mathcal{U}$. If this property holds for an arbitrary family of shrunk sets in $\mathcal{U}$, we say that $(\hat{G}, \mathcal{U}, t)$ *admits expansion*. This is exactly the class of $(G, \mathcal{U}, t)$ to which our algorithm is applicable.

Definition 4 *Let $\hat{G} = (\hat{V}, \hat{E})$ be a bipartite graph, $\mathcal{U} \subseteq 2^{\hat{V}}$, and t be a positive integer. For arbitrary $U_1, \ldots, U_l \in \mathcal{U}$ that are pairwise disjoint, let $G = (V, E)$ denote the graph obtained from $\hat{G}$ by executing* SHRINK(U_1), *...,* SHRINK(U_l), *and let $F \subseteq E$ be an arbitrary edge set satisfying* (4). *We say that* $(\hat{G}, \mathcal{U}, t)$ admits expansion *if there exists $F_{U_i} \subseteq \hat{E}[U_i]$ satisfying $|F_{U_i}| = \lfloor (t|U_i|-1)/2 \rfloor$ and* (5) *for each $i = 1, \ldots, l$.*

In what follows we assume that $(\hat{G}, \mathcal{U}, t)$ admits expansion. Now this assumption and the degree constraint (4) guarantee that we can always obtain a t-matching $\hat{F} = F \cup \bigcup\{F_U \mid U \in \mathcal{U}$ is a maximal shrunk set$\}$ in $\hat{G}$.

Furthermore, we should take $\mathcal{U}$-feasibility of $\hat{F}$ into account. We refer to F in G as *feasible* if $\hat{F}$ is $\mathcal{U}$-feasible. If there are several possibilities of F_U for shrunk $U \in \hat{U}$, we say that F is $\mathcal{U}$-feasibleif there is at least one $\hat{F}$ which is $\mathcal{U}$-feasible. In other words, if F satisfying (4) is not feasible, then there exists $U \in \mathcal{U}$ such that

$$\deg_F(v) = \begin{cases} t & (v \in U_{\rm n}), \\ 1 & (v \in U_{\rm p}), \end{cases} \tag{6}$$

and $\hat{F}$ shall have a t-factor in $\hat{G}[U]$.

We are now ready for the entire description of our algorithm. The algorithm begins with $G = \hat{G}$ and an arbitrary $\mathcal{U}$-feasible t-matching $F \subseteq \hat{E}$, typically $F = \emptyset$. We first construct an auxiliary digraph.

Procedure AuxiliaryDigraph(G, F)**.** Construct a digraph (V, A) defined by

$$A = \{(u, v) \mid u \in V^+,\, v \in V^-,\, \{u, v\} \in E \setminus F\} \cup \{(v, u) \mid u \in V^+,\, v \in V^-,\, \{u, v\} \in F\}.$$

Define the sets of source vertices $S \subseteq V^+$ and sink vertices $T \subseteq V^-$ by

$$S = \{u \in V_{\rm n}^+ \mid \deg_F(u) \leq t-1\} \cup \{u_U^+ \in V_{\rm p}^+ \mid \deg_F(u_U^+) = 0\},$$
$$T = \{v \in V_{\rm n}^- \mid \deg_F(v) \leq t-1\} \cup \{v_U^- \in V_{\rm p}^- \mid \deg_F(v_U^-) = 0\}.$$

Now return $D = (V, A; S, T)$.

Suppose that there exists a directed path $P = (e_1, f_1, \ldots, e_l, f_l, e_{l+1})$ in D from S to T. Note that $e_i \in E \setminus F$ $(i = 1, \ldots, l+1)$ and $f_i \in F$ $(i = 1, \ldots, l)$. Denote the symmetric difference $(F \setminus P) \cup (P \setminus F)$ of F and P by $F \triangle P$. If $F \triangle P$ is feasible, we execute AUGMENT(G, F, P) below, and then EXPAND(G, F).

Procedure Augment(G, F, P)**.** Let $F := F \triangle P$ and return F.

If $F \triangle P$ is not feasible, we apply SHRINK(U) after determining a set $U \in \mathcal{U}$ to be shrunk by the following procedure.

Procedure ViolatingSet(G, F, P)**.** For $j = 1, \ldots, l$, define $F_j = (F \setminus \{f_1, \ldots f_j\}) \cup \{e_1, \ldots, e_j\}$. Also define $F_0 = F$ and $F_{l+1} = F \triangle P$. Let j^* be the minimum index j such that F_j is not feasible, let $U \in \mathcal{U}$ be an arbitrary set satisfying (6) for $F = F_{j^*}$. Now let $F := F_{j^*-1}$, and return (F, U).

Finally, if D does not have a directed path from S to T, we determine $X \subseteq \hat{V}$ minimizing the right-hand side of (1) as follows.

Procedure Minimizer(G, F)**.** Let $R \subseteq V$ be the set of vertices reachable from S, and $X := (V^+ \setminus R^+) \cup R^-$. If a natural vertex $v \in V^- \setminus X$ has t edges in F connecting R^+ and v, then $X := X \cup \{v\}$. If a pseudovertex $v_U^- \in V^- \setminus X$ has one edge in F connecting R^+ and v_U^-, then $X := X \cup \{v_U^-\}$. Finally, return $X := \hat{X}$.

We then apply EXPAND(G, F) and the algorithm terminates by returning $F \subseteq \hat{E}$ and $X \subseteq \hat{V}$.

Now the description of the algorithm is completed. The optimality of F and X will be proved in Sect. 3.3.

3.3 Min-max Theorem: Strong Duality

In this section, we strengthen Lemma 3 to be a min-max relation and prove the validity of our algorithm in Sect. 3.2. That is, we show that the output (F, X) of the algorithm satisfies (1) with equality. This constructively proves the following min-max relation for the class of $(G, \mathcal{U}, t)$ admitting expansion.

Theorem 5 *Let $G = (V, E)$ be a bipartite graph, $\mathcal{U} \subseteq 2^V$, and t be a positive integer such that $(G, \mathcal{U}, t)$ admits expansion. Then, the maximum size of a $\mathcal{U}$-feasible t-matching is equal to the minimum of*

$$t|X| + |E[C_{V \setminus X}]| + \sum_{U \in \mathcal{U}_{V \setminus X}} \left\lfloor \frac{t|U| - 1}{2} \right\rfloor,$$

where X runs over all subsets of V.

PROOF: It suffices to prove that (2) and (3) hold by equality for the output $(\hat{F}, \hat{X})$ of the algorithm.

First, since X is defined based on reachability in the auxiliary digraph D, $F[X] = \emptyset$ holds when no directed path from S to T is found. Moreover, it is not difficult to see that $v_U^+ \in R$ holds for every pseudovertex v_U^+. Hence it follows that $\hat{F}[\hat{X}] = \emptyset$.

Second, for every $v \in \hat{X}$, $\deg_{\hat{F}}(v) = t$ holds, and thus (2) holds by equality.

Finally, edges in $\hat{G}[\hat{V} \setminus \hat{X}]$ are in F before the last EXPAND(G, F) or obtained by expanding pseudovertices u_U^+ and v_U^-, which are isolated vertices in $G[V \setminus X]$. This means that U forms a component in $\hat{G}[\hat{V} \setminus \hat{X}]$, and thus the equality in (3) follows. □

4 Weighted $\mathcal{U}$-feasible t-matching

In this section, we extend the min-max theorem and the algorithm presented in Sect. 3 to the maximum-weight $\mathcal{U}$-feasible t-matching problem. Recall that G is a bipartite graph in which every edge may have parallel copies with the same weight, and $(G, \mathcal{U}, t)$ admits expansion. We assume that w is vertex-induced on each $U \in \mathcal{U}$, which commonly extends the assumptions for the maximum-weight square-free and even factor problems.

4.1 Linear Program

Described below is a linear programming relaxation of the the maximum-weight $\mathcal{U}$-feasible t-matching problem, where the variable is $x \in \mathbf{R}^E$:

$$\begin{aligned}
\text{(P)}\quad \text{maximize} \quad & \sum_{e \in E} w(e)x(e) & \\
\text{subject to} \quad & x(\delta(v)) \le t & (v \in V),\\
& x(E[U]) \le \left\lfloor \frac{t|U|-1}{2} \right\rfloor & (U \in \mathcal{U}),\\
& 0 \le x(e) \le 1 & (e \in E).
\end{aligned}$$

We shall remark that the second constraint, describing $\mathcal{U}$-feasibility, is a common extension of the blossom constraint for the nonbipartite matching problem (put $t = 1$), and the subtour elimination constraints for the TSP (put $t = 2$).

Its dual program, where the variables are $p \in \mathbf{R}^V$, $q \in \mathbf{R}^E$, and $r \in \mathbf{R}^{\mathcal{U}}$, is given as follows:

$$\begin{aligned}
\text{(D)}\quad \text{minimize} \quad & t\sum_{v \in V} p(v) + \sum_{e \in E} q(e) + \sum_{U \in \mathcal{U}} \left\lfloor \frac{t|U|-1}{2} \right\rfloor r(U) & \\
\text{subject to} \quad & p(u) + p(v) + q(e) + \sum_{U \in \mathcal{U}:\ e \in E[U]} r(U) \ge w(e) & (e = \{u,v\} \in E),\\
& p(v) \ge 0 & (v \in V),\\
& q(e) \ge 0 & (e \in E),\\
& r(U) \ge 0 & (U \in \mathcal{U}).
\end{aligned}$$

Define $w' \in \mathbf{R}^E$ by $w'(e) = p(u) + p(v) + q(e) + \sum_{U \in \mathcal{U}:\ e \in E[U]} r(U) - w(e)$ for $e = \{u,v\} \in E$. The complementary slackness conditions for (P) and (D) are as follows.

$$x(e) > 0 \Longrightarrow w'(e) = 0 \qquad (e \in E), \tag{7}$$

$$p(v) > 0 \Longrightarrow x(\delta(v)) = t \qquad (v \in V), \tag{8}$$

$$q(e) > 0 \Longrightarrow x(e) = 1 \qquad (e \in E), \tag{9}$$

$$r(U) > 0 \Longrightarrow x(E[U]) = \left\lfloor \frac{t|U|-1}{2} \right\rfloor \qquad (U \in \mathcal{U}). \tag{10}$$

4.2 Primal-dual Algorithm

In this section, we exhibit a combinatorial primal-dual algorithm for the maximum-weight $\mathcal{U}$-feasible t-matching problem in bipartite graphs, where $(G, \mathcal{U}, t)$ admits expansion and w is vertex-induced for each $U \in \mathcal{U}$.

We maintain primal and dual feasible solutions satisfying (7), (9), (10), and (8) for $v \in V^-$. The algorithm terminates when (8) is attained for every $v \in V^+$. Again denote the input graph by $\hat{G} = (\hat{V}, \hat{E})$, and the graph in hand, i.e., the graph resulted from possibly repeated shrinkings, by $G = (V, E)$. The variables in the algorithm are $F \subseteq E$, $p \in \mathbf{R}^{\hat{V}}$, $q \in \mathbf{R}^{\hat{E}}$, and $r \in \mathbf{R}^{\mathcal{U}}$. Note that p and q are always defined on the original vertex and edge sets, respectively.

In the beginning, we set

$$\begin{aligned}
& F = \emptyset, && p(v) = \begin{cases} \max\{w(e) \mid e \in \delta(v)\} & (v \in V^+),\\ 0 & (v \in V^-), \end{cases}\\
& q(e) = 0 \quad (e \in E), && r(U) = 0 \quad (U \in \mathcal{U}).
\end{aligned} \tag{11}$$

The auxiliary digraph D is constructed as follows. Major differences from Sect. 3.2 are that we only use an edge e with $w'(e) = 0$, and a vertex in V^+ can become a sink vertex.

Procedure AuxiliaryDigraph(G, F, p, q, r)**.** Define a digraph (V, A) by

$$A = \{(\partial^+ e, \partial^- e) \mid e \in E \setminus F,\ w'(e) = 0\} \cup \{(\partial^- e, \partial^+ e) \mid e \in F\}.$$

The sets of source vertices $S \subseteq V^+$ and sink vertices $T \subseteq V^+ \cup V^-$ are defined by

$$\begin{aligned} S &= \{u \in V^+_{\rm n} \mid \deg_F(v) \leq t-1,\ p(u) > 0\} \cup \{u^+_U \in V^+_{\rm p} \mid \deg_F(u^+_U) = 0,\ p(u) > 0 \text{ for some } u \in U\} \\ T &= \{v \in V^-_{\rm n} \mid \deg_F(v) \leq t-1\} \cup \{v^-_U \in V^-_{\rm p} \mid \deg_F(v^-_U) = 0\} \\ &\quad \cup \{u \in V^+_{\rm n} \mid \deg_F(u) = t,\ p(u) = 0\} \cup \{u^+_U \in V^+_{\rm p} \mid \deg_F(u^+_U) = 1,\ p(u) = 0 \text{ for some } u \in U\}. \end{aligned}$$

Return $D = (V, A; S, T)$,

Suppose that D has a directed path P from S to T, and let $F' := F \triangle P$.

If F' is feasible, we execute AUGMENT(G, F, P), which is the same as in Sect. 3.2. Note that, if P ends in a vertex in $T \cap V^+$, then $|F|$ does not increase. However, in this case the number of vertices satisfying (8) increases by one, and we get closer to the termination condition ((8) for every vertex).

If F' is not feasible, apply VIOLATINGSET(G, F, P) as in Sect. 3.2. For the output U of VIOLATINGSET(G, F, P), execute MODIFY(G, F, U) below if $p(u) = 0$ for some $u \in U^+$. Otherwise apply SHRINK(U).

Procedure Modify(G, F, U)**.** Let $u^* \in U^+$ satisfy $p(u^*) = 0$. Then find $K \subseteq E[U]$ such that

$$\deg_K(u) = \begin{cases} t & (u \in U^+_{\rm n} \setminus \{u^*\}), \\ t-1 & (u = u^*), \\ 0 & (u = u^+_{U'} \in U^+_{\rm p}, u^* \in U'), \\ \deg_{F[U]}(u) & (u \in U^-_{\rm n} \cup U^-_{\rm p}). \end{cases}$$

Now return $F := (F \setminus F[U]) \cup K$.

If D does not have a directed path from S to T, then update the dual variables p, q, and r by procedure DUALUPDATE(G, F, p, q, r) described below.

Procedure DualUpdate(G, F, p, q, r)**.** Let $R \subseteq V$ be the set of vertices reachable from S in the auxiliary digraph D. Then,

$$p(v) := \begin{cases} p(v) - \epsilon & (v \in \hat{R}^+), \\ p(v) + \epsilon & (v \in \hat{R}^-), \\ p(v) & (v \in \hat{V} \setminus \hat{R}), \end{cases}$$

$$q(e) := \begin{cases} q(e) + \epsilon & (\partial^+ e \in \hat{R}^+,\ \partial^- e \in \hat{V}^- \setminus \hat{R}^-), \\ q(e) & (v \in \hat{V}^- \setminus \hat{R}^-), \end{cases}$$

$$r(U) := \begin{cases} r(U) + \epsilon & (u^+_U \in R^+, v^-_U \in V^- \setminus R^-), \\ r(U) - \epsilon & (u^+_U \in V^+ \setminus R^+, v^-_U \in R^+), \\ r(U) & (\text{otherwise}), \end{cases}$$

where

$$\epsilon = \min\{\epsilon_1, \epsilon_2, \epsilon_3\}, \quad \epsilon_1 = \min\{w'(\{u, v\}) \mid u \in \hat{R}^+, v \in \hat{V}^- \setminus \hat{R}^-\},$$
$$\epsilon_2 = \min\{p(u) \mid u \in \hat{R^+}\}, \quad \epsilon_3 = \min\{r(U) \mid u^+_U \in \hat{V}^+ \setminus \hat{R}^+, v^-_U \in \hat{R}^-\}.$$

Then return (p, q, r).

Finally, we expand every U satisfying $r(U) = 0$ after AUGMENT(G, F, P), MODIFY(G, F, U), and DUALUPDATE(G, F, p, q, r). We note that, if any $U' \subsetneq U$ satisfies $r_{U'} > 0$, which implies that U' had been shrunk before U was shrunk, then U' is maintained to be shrunk.

Procedure Expand(G, F, r)**.** For each shrunk $U \in \mathcal{U}$ with $r(U) = 0$, execute the following procedures. Update G by replacing u_U^+ and v_U^- by the graph induced by $U_{\rm n} \cup U_{\rm p}$ just before SHRINK(U) is applied. Determine $F_U \subseteq E[U_{\rm n} \cup U_{\rm p}]$ of $(t|U_{\rm n}| + |U_{\rm p}|)/2 - 1$ edges so that $F' = F \cup F_U$ can be extended to a $\mathcal{U}$-feasible t-matching in $\hat{G}$. Then return $F := F'$.

The algorithm constructively proves the following theorem for the integrality of (P) and (D). This is a common extension of dual integrality theorems for nonbipartite matchings [6], even factors [14], triangle-free 2-matchings [3], and square-free 2-matchings [18].

Theorem 6 *If $(G,\mathcal{U},t)$ admits expansion and w is vertex-induced on each $U \in \mathcal{U}$, then the linear program* (P) *has an integer optimal solution. Moreover, the dual program* (D) *also has an integer optimal solution such that $\{U \in \mathcal{U} \mid r(U) > 0\}$ is a laminar family.*

5 Conclusion

We have presented a new framework of the optimal $\mathcal{U}$-feasible t-matching problem. Then we have established a min-max theorem and a combinatorial algorithm under the reasonable assumption that G is bipartite, $(G,\mathcal{U},t)$ admits expansion, and w is vertex-induced on each $U \in \mathcal{U}$. Our problem under these assumptions can describe a number of generalizations of the matching problem, such as the matching and triangle-free 2-matching problem in nonbipartite graphs, and the square-free 2-matching problem in bipartite graphs. We have also seen that $\mathcal{U}$-feasibility is a common generalization of the blossom constraints for the nonbipartite matching problem and the subtour elimination constraints for the TSP. We anticipate that this unified perspective provides a new approach to the TSP utilizing matching theory.

Acknowledgements

The author is obliged to Yutaro Yamaguchi for the helpful comments on the draft of the paper.

References

[1] K. Bérczi and L.A. Végh, Restricted b-matchings in degree-bounded graphs, in F. Eisenbrand and B. Shepherd, eds., *Integer Programming and Combinatorial Optimization: Proceedings of the 14th IPCO, LNCS* **6080** (2010), Springer-Verlag, 43–56

[2] S. Boyd, S. Iwata and K. Takazawa, Finding 2-factors closer to TSP tours in cubic graphs, *SIAM Journal on Discrete Mathematics* **27** (2013), 918–939

[3] G. Cornuéjols and W. Pulleyblank, A matching problem with side conditions, *Discrete Mathematics* **29** (1980), 135–159

[4] W.H. Cunningham and J.F. Geelen, The optimal path-matching problem, *Combinatorica* **17** (1997), 315–337

[5] W.H. Cunningham and J.F. Geelen, Vertex-disjoint dipaths and even dicircuits, unpublished manuscript, 2001

[6] W.H. Cunningham and A.B. Marsh, III, A primal algorithm for optimum matching, *Mathematical Programming Study* **8** (1978), 50–72

[7] J. Edmonds, Paths, trees, and flowers, *Canadian Journal of Mathematics*, **17** (1965), 449–467

[8] A. Frank, Restricted t-matchings in bipartite graphs, *Discrete Applied Mathematics* **131** (2003), 337–346

[9] D. Hartvigsen, *Extensions of Matching Theory*, Ph.D. thesis, Carnegie Mellon University, 1984

[10] D. Hartvigsen, Finding maximum square-free 2-matchings in bipartite graphs, *Journal of Combinatorial Theory* **B96** (2006), 693–705

[11] S. Iwata and K. Takazawa, The independent even factor problem, *SIAM Journal on Discrete Mathematics*, **22** (2008), 1411–1427

[12] T. Kaiser and R. Škrekovski, Cycles intersecting edge-cuts of prescribed sizes, *SIAM Journal on Discrete Mathematics* **22** (2008), 861–874

[13] Z. Király, C_4-free 2-factors in bipartite graphs, EGRES Technical Report TR-2001-13, 1999

[14] T. Király and M. Makai, On polyhedra related to even factors, in D. Bienstock and G.L. Nemhauser, eds., *Integer Programming and Combinatorial Optimization: Proceedings of the 10th IPCO, LNCS* **3064** (2004), Springer-Verlag, 416–430

[15] Y. Kobayashi, J. Szabó and K. Takazawa, A proof of Cunningham's conjecture on restricted subgraphs and jump systems, *Journal of Combinatorial Theory* **B102** (2012), 948–966

[16] Y. Kobayashi and K. Takazawa, Even factors, jump systems, and discrete convexity, *Journal of Combinatorial Theory* **B99** (2009), 139–161

[17] L. Lovász and M.D. Plummer, *Matching Theory*, AMS Chelsea Publishing, Providence, 2009

[18] M. Makai, On maximum cost $K_{t,t}$-free t-matchings of bipartite graphs, *SIAM Journal on Discrete Mathematics* **21** (2007), 349–360

[19] G. Pap, A TDI description of restricted 2-matching polytopes, in D. Bienstock and G.L. Nemhauser, eds., *Integer Programming and Combinatorial Optimization: Proceedings of the 10th IPCO, LNCS* **3064** (2004), Springer-Verlag, 139–151

[20] G. Pap, Combinatorial algorithms for matchings, even factors and square-free 2-factors, *Mathematical Programming* **110** (2007), 57–69

[21] G. Pap and L. Szegő, On the maximum even factor in weakly symmetric graphs, *Journal on Combinatorial Theory* **B91** (2004), 201–213

[22] K. Takazawa, A weighted even factor algorithm, *Mathematical Programming* **22** (2008), 223–237

[23] K. Takazawa, A weighted $K_{t,t}$-free t-factor algorithm for bipartite graphs, *Mathematics of Operations Research* **34** (2009), 351–362

[24] K. Takazawa, Decomposition theorems for square-free 2-matchings in bipartite graphs, in E.W. Mayr, ed., *Graph-Theoretic Concepts in Computer Science: Proceedings of the 41st International Workshop WG 2015, LNCS* **9224** (2016), Springer-Verlag, 373–387

[25] K. Takazawa, Finding a maximum 2-matching excluding prescribed cycles in bipartite graphs, in P. Faliszewski, A. Muscholl and R. Niedermeier, eds., *the 41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, LIPIcs* **58** (2016), 87:1–87:14

Nash Equilibria in Combinatorial Auctions with Item Bidding and Subadditive Valuations

Hiroyuki Umeda

Information and System Engineering
Graduate School of Science and Engineering
Chuo University
Tokyo, Japan
a16.x6wr@g.chuo-u.ac.jp

Takao Asano [1]

Information and System Engineering
Faculty of Science and Engineering
Chuo University
Tokyo, Japan
asano@ise.chuo-u.ac.jp

Abstract: We discuss Nash equilibria in combinatorial auctions with item bidding. Specifically, we give a characterization for the existence of a Nash equilibrium in such a combinatorial auction when valuations by n bidders satisfy symmetric and subadditive properties. By this characterization, we can obtain an algorithm for deciding whether a Nash equilibrium exists in such a combinatorial auction.

Keywords: Nash equilibrium, combinatorial auction, price of anarchy, social welfare problem, subadditivity

1 Introduction

In a *combinatorial auction*, m items $M = \{1, 2, \ldots, m\}$ are offered for sale to n bidders $N = \{1, 2, \ldots, n\}$. Each bidder i has a valuation f_i that assigns nonnegative number to every subset S of M. The objective is to find a partition $S_1, S_2, \ldots, S_n$ of M such that the *social welfare* $\sum_{i=1}^{n} f_i(S_i)$ is maximized. The combinatorial auction problem is sometimes called the *social welfare problem* when we disregard strategic issues on bidders' selfish concerns. VCG (Vickrey-Clarke-Groves) mechanisms optimize the social welfare in a combinatorial auction with selfish bidders. However, it may take exponential time in m and n. Actually, the social welfare problem is shown to be NP-hard by Lehmann, Lehmann and Nisan, even if every valuation f_i ($i \in N$) satisfies submodularity [12] ($f_i : 2^M \to \mathbf{R}_+$ is *submodular* if $f_i(S \cup T) + f_i(S \cap T) \leq f_i(S) + f_i(T)$ for all $S, T \subseteq M$ and is *subadditive* if $f_i(S \cup T) \leq f_i(S) + f_i(T)$ for all $S, T \subseteq M$). Therefore approximation algorithms have been proposed for the social welfare problem. Since each valuation f_i is defined by 2^m subsets of M, most approximation algorithms are based on oracle models. Two oracle models, the value queries oracle model and the demand queries oracle model, are commonly used. Furthermore, in most proposed approximation algorithms, each valuation f_i is restricted to satisfy some conditions. Two restrictions, submodularity and subadditivity, are commonly used.

For the submodular social welfare problem (i.e., each valuation is submodular) with the value queries oracle model, the following are known. Lehmann, Lehmann and Nisan proposed a $\frac{1}{2}$-approximation algorithm [12]. Khot et al. showed that this problem cannot be approximated to a factor better than $1 - \frac{1}{e}$ unless **P**= **NP** [10], where e is the base of the natural logarithm. Vondrák proposed a randomized $(1 - \frac{1}{e})$-approximation algorithm [15]. Using the more powerful demand queries oracle model, Dobzinski and Schapira proposed an improved $(1 - \frac{1}{e})$-approximation algorithm for the submodular social welfare problem [6].

For the more general subadditive social welfare problem (where each valuation is subadditive), Dobzinski, Nisan, and Schapira proposed an $\Omega(1/\log m)$-approximation algorithm using the value queries oracle

[1] Research is supported by Grants-in-Aid for Challenging Exploratory Research (15K11988) and Chuo University Personal Research Grant.

model [5]. Using the more powerful demand queries oracle model, Feige proposed a $\frac{1}{2}$-approximation algorithm for the subadditive social welfare problem and also showed that it is NP-hard to approximate to a factor better than $\frac{1}{2}$ [8]. He also proposed a $(1-\frac{1}{e})$-approximation algorithm for the fractional subadditive (more general than submodular, but more restricted than subadditive) social welfare problem.

For a partition $S_1, \ldots, S_n$ of M in a combinatorial auction, where each bidder i obtains the items in S_i, the price, denoted by $price(S_i)$, is attached to S_i. The *payoff* of bidder i is defined by $f_i(S_i) - price(S_i)$. Each selfish bidder i wants to maximize his payoff. The combinatorial auctions that are used in practice are different from VCG mechanisms. For example, eBay uses an auction in which m items are sold in m independent second-price auctions. Thus, item bidding, as a combinatorial auction scheme, occurs rather "spontaneously" and this type of auction is called a *combinatorial auction with item bidding* [3]. Thus, a bidder's strategy is the m-dimensional vector of his bids he submits in the different single-item auctions. A bid profile of all bidders' bid vectors is a *pure Nash equilibrium* if no bidder wants to change his bid vector assuming that any other bidders keep their own bid vectors.

For a combinatorial auction with item bidding in which all bidders' valuations are submodular, Christodoulou, Kovács, and Schapira showed that there is always a pure Nash equilibrium and proposed an algorithm for finding a pure Nash equilibrium which is a $\frac{1}{2}$-approximation to the optimal social welfare in polynomial time in n and m [3]. Bhawalkar and Roughgarden considered a combinatorial auction with item bidding where all bidders' valuations are subadditive and showed that every pure Nash equilibrium has a welfare at least $\frac{1}{2}$ of social optimal welfare (thus, the *price of anarchy*, the ratio of the social optimal welfare to the welfare of the worst Nash equilibrium, is at most 2) under the assumption of no "overbidding" [1]. Furthermore, Bhawalkar and Roughgarden suggested the following open problem: "Identify necessary and sufficient conditions for the existence of a pure Nash equilibrium in a combinatorial auction with item bidding and subadditive valuations."

In this paper, we give a necessary and sufficient condition for the existence of a pure Nash equilibrium in a combinatorial auction with item bidding and subadditive valuations when valuations are *symmetric* (i.e., $f_i(S) = f_i(T)$ for all subsets $S, T \subseteq M$ with $|S| = |T|$) under the assumption of no "overbidding". Symmetric valuations were considered in [12, 13]. An auction with symmetric valuations is called a *multi-unit auction* and several results were obtained in multi-unit auctions [2, 9, 11]. The auction for the super-long-term Japanese Government Bonds is an example of multi-unit auctions [7].

2 Combinatorial auctions and item bidding

In a combinatorial auction we are given a set of n bidders $N = \{1, 2, \ldots, n\}$ and a set of m items $M = \{1, 2, \ldots, m\}$. Each bidder $i \in N$ has a valuation f_i which assigns, for each subset $S \subseteq M$, a nonnegative value $f_i(S)$. We denote a valuation profile of n bidders by $\boldsymbol{f} = (f_1, f_2, \ldots, f_n)$. In a combinatorial auction with item bidding, each bidder $i \in N$ has a nonnegative bid $b_i(j)$ for each item $j \in M$ and i's bid is denoted by $b_i = (b_i(1), b_i(2), \ldots, b_i(m))$. We denote a bid profile of n bidders by $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$. We also write $\boldsymbol{b}_{-i} = (b_1, b_2, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$ for each $i \in N$ which is obtained by deleting i's bid b_i from $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$. For $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ and for each bidder $i \in N$ and each item $j \in M$, we denote by $b_{-i}^{\max}(j)$ the maximum bid among the bids other than i's bid, i.e., $b_{-i}^{\max}(j) = \max_{h \in N - \{i\}}\{b_h(j)\}$. For each bidder $i \in N$, we write

$$b_{-i}^{\max} = (b_{-i}^{\max}(1), b_{-i}^{\max}(2), \ldots, b_{-i}^{\max}(m)). \tag{1}$$

Feasibility of $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ (i.e., "no overbidding") is defined as follows.

Definition 1 *For each $i \in N$, if there is a subset $S \subseteq M$ such that $\sum_{j \in S} b_i(j) > f_i(S)$ then bid b_i is called overbidding. Otherwise (i.e., $\sum_{j \in S} b_i(j) \leq f_i(S)$ for all $S \subseteq M$), b_i is feasible. If all b_i $(i \in N)$ are feasible, then bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is called feasible.*

In a combinatorial auction with item bidding [1],[3], the second price auction is used. Thus, items are allocated as follows. In a bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$, if bidder $i \in N$ has bid $b_i(j)$ for $j \in M$

which is higher than any other bidders' bids $b_h(j)$ ($h \in N - \{i\}$), then item j is allocated to i. That is, if $b_i(j) > b_{-i}^{\max}(j)$ then bidder i will win and obtain $j \in M$. In this case, the *price* of item $j \in M$, denoted by $price(j)$, is defined by the second highest bid among the bids of all bidders. Thus, $price(j) = b_{-i}^{\max}(j) = \max\{b_h(j) \mid h \in N - \{i\}\}$. This implies that bidder $i \in N$ can obtain no item $j \in M$ with $b_i(j) < b_{-i}^{\max}(j)$.

For item $j \in M$, if the highest positive bid for j is attained by two or more bidders, then exactly one of such bidders will win and obtain j. In this case, if i wins, then the price of j will be $price(j) = b_{-i}^{\max}(j) = b_i(j)$. In this paper, we assume that, for each item $j \in M$, some bidder's bid is positive. For a bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ and for each bidder $i \in N$, let $X_i(\boldsymbol{b})$ be the set of items (i wins and) allocated to i. Then $X_i(\boldsymbol{b}) \subseteq \{j \in M \,|\, b_i(j) = \max\{b_h(j) \,|\, h \in N\}\}$ by the argument above. The *payoff* $u_i(X_i(\boldsymbol{b}))$ of bidder $i \in N$ for $X_i(\boldsymbol{b})$ is defined by $u_i(X_i(\boldsymbol{b})) = f_i(X_i(\boldsymbol{b})) - \sum_{j \in X_i(\boldsymbol{b})} price(j)$.

Nash equilibrium is defined as follows. For a feasible bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$, let $X_i(\boldsymbol{b})$ be the set of items allocated to bidder i. Suppose that only bidder $i \in N$ changes bid b_i to b_i' and let $\boldsymbol{b}_i' = (b_1, b_2, \ldots, b_{i-1}, b_i', b_{i+1}, \ldots, b_n)$ be the resultant bid profile of all bidders. For convenience, we sometimes write $(b_i', \boldsymbol{b}_{-i})$ in place of $\boldsymbol{b}_i' = (b_1, b_2, \ldots, b_{i-1}, b_i', b_{i+1}, \ldots, b_n)$. Furthermore, let $X_i(b_i', \boldsymbol{b}_{-i})$ be the set of items allocated to i in bid profile $(b_i', \boldsymbol{b}_{-i})$. Suppose that, even if bidder i changes bid b_i to arbitrary feasible bid b_i', the i's payoff $u(X_i(b_i', \boldsymbol{b}_{-i}))$ will not become strictly higher than $u_i(X_i(\boldsymbol{b}))$. In this case, i does not want to change the bid b_i in $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$. If no bidder $i \in N$ wants to change the bid b_i in the feasible bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$, that is, if $u_i(X_i(\boldsymbol{b})) \geq u_i(X_i(b_i', \boldsymbol{b}_{-i}))$ for all bidders $i \in N$ and for all feasible bid profiles $(b_i', \boldsymbol{b}_{-i})$ (and $X_i(b_i', \boldsymbol{b}_{-i})$) defined above, then $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is called a *pure Nash equilibrium* (*Nash equilibrium* in short).

In this paper, we make the following assumptions on each valuation f_i ($i \in N$): (i) $f_i(\emptyset) = 0$ (normalization), (ii) $0 < f_i(S) \leq f_i(T)$ for all subsets $S, T \subseteq M$ with $\emptyset \neq S \subset T$ (monotonicity), (iii) $f_i(S \cup T) \leq f_i(S) + f_i(T)$ for all subsets $S, T \subseteq M$ (subadditivity), and (iv) $f_i(S) = f_i(T)$ for all subsets $S, T \subseteq M$ with $|S| = |T|$ (*symmetry*). Thus, we can define $v_i : \{0, 1, 2, \ldots, m\} \to \mathbf{R}_+$ by $v_i(|S|) = f_i(S)$ for any subset $S \subseteq M$. Then v_i is well defined by symmetry of f_i in the assumption above. Using this symmetric valuation v_i, we can write (i), (ii) and (iii) in the assumption above and the payoff as follows.

Definition 2 *Each v_i ($i \in N$) in $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ satisfies the following:*

1. *(Normalization) $v_i(0) = 0$.*
2. *(Monotonicity) $0 < v_i(k) \leq v_i(k')$ for all k, k' with $1 \leq k < k' \leq m$.*
3. *(Subadditivity) $v_i(\min\{k + k', m\}) \leq v_i(k) + v_i(k')$ for all k, k' with $1 \leq k, k' \leq m$.*

Definition 3 *The payoff $u_i(X_i(\boldsymbol{b}))$ of bidder i is written by*

$$u_i(X_i(\boldsymbol{b})) = v_i(|X_i(\boldsymbol{b})|) - \sum_{j \in X_i(\boldsymbol{b})} price(j). \tag{2}$$

Since we will give a characterization of the existence of Nash equilibria under the assumption of no "overbidding", we first consider the feasibility of bids.

Definition 4 *For each bidder $i \in N$, let v_i be a valuation satisfying Definition 2 and let w_i be a function with $w_i(0) = 0$ and, for each $k_i \in \{1, 2, \ldots, m\}$,*

$$w_i(k_i) = k_i \min\left\{v_i(1), \frac{v_i(2)}{2}, \ldots, \frac{v_i(k_i - 1)}{k_i - 1}, \frac{v_i(k_i)}{k_i}\right\}. \tag{3}$$

From now on, we assume that each v_i ($i \in N$) satisfies Definition 2 and each w_i ($i \in N$) is the function defined in Definition 4. Then we have the following lemma and theorem (see Appendix for their proofs). They will play a central role in the proof of the main results.

Lemma 5 *Each w_i $(i \in N)$ has the following properties:*

$$w_i(0) = v_i(0), \quad w_i(1) = v_i(1), \quad w_i(k_i) \le v_i(k_i) \quad (k_i = 2, 3, \ldots, m), \tag{4}$$

$$w_i(k_i) = k_i \min\left\{ \frac{w_i(k_i - 1)}{k_i - 1}, \frac{v_i(k_i)}{k_i} \right\} \quad (k_i = 2, 3, \ldots, m), \tag{5}$$

$$w_i(1) \ge \frac{w_i(2)}{2} \ge \cdots \ge \frac{w_i(m)}{m}, \quad w_i(1) \le w_i(2) \le \cdots \le w_i(m), \quad \textit{and} \tag{6}$$

$$\textit{if } w_i(k_i) < v_i(k_i) \textit{ then } w_i(k_i) = \frac{k_i}{k_i - 1} w_i(k_i - 1) \quad (k_i = 2, 3, \ldots, m). \tag{7}$$

Theorem 6 *For any bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ and for each bidder $i \in N$, let the elements of each $b_i = (b_i(1), b_i(2), \ldots, b_i(m))$ $(i \in N)$ be ordered in nondecreasing order by using a permutation π_i on $M = \{1, 2, \ldots, m\}$ as follows:*

$$b_i(\pi_i(1)) \le b_i(\pi_i(2)) \le \cdots \le b_i(\pi_i(m)). \tag{8}$$

Then bidder i's bid $b_i = (b_i(1), b_i(2), \ldots, b_i(m))$ is feasible if and only if

$$\sum_{j=m-k_i+1}^{m} b_i(\pi_i(j)) \le w_i(k_i) \quad (k_i = 1, 2, \ldots, m), \tag{9}$$

that is, the sum of largest k_i bids in $b_i = (b_i(1), b_i(2), \ldots, b_i(m))$ is at most $w_i(k_i)$ for all $k_i \in \{1, 2, \ldots, m\}$. Thus, the bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is feasible if and only if $\sum_{j=m-k_i+1}^{m} b_i(\pi_i(j)) \le w_i(k_i)$ hold for all $i \in N$ and for all $k_i \in \{1, 2, \ldots, m\}$.

By Theorem 6, we have the following corollary.

Corollary 7 *A bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ of n bidders can be determined whether it is feasible or not in $O(mn)$ time, if the elements of b_i for all $i \in N$ have been sorted as in (8).*

3 Existence of Nash equilibria

In this section, we first give some terms and lemmas for explaining the main result in this paper, and then give an outline of its proof.

Let $P = (M_1, M_2, \ldots, M_n)$ be a partition of M into n subsets, i.e.,

$$M_i \cap M_h = \emptyset \quad (i, h \in N, i \ne h), \qquad M_1 \cup M_2 \cup \cdots \cup M_n = M. \tag{10}$$

For each $i \in N$, let $d_i = (d_i(1), d_i(2), \ldots, d_i(m))$ be defined by

$$d_i(j) = \begin{cases} \frac{w_i(|M_i|)}{|M_i|} & (j \in M_i), \\ 0 & (j \in M - M_i). \end{cases} \tag{11}$$

Then we have the following lemma (see Appendix for its proof) and the main result.

Lemma 8 *The bid profile $\boldsymbol{d} = (d_1, d_2, \ldots, d_n)$ defined by Equation (11) is feasible and $X_i(\boldsymbol{d}) = M_i$ for each $i \in N$ (i.e., the set of items allocated to i in $\boldsymbol{d} = (d_1, d_2, \ldots, d_n)$ is M_i).*

Theorem 9 *A valuation profile $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ satisfying Definition 2 has a Nash equilibrium if and only if there is a partition $P = (M_1, M_2, \ldots, M_n)$ of M into n subsets of such that the feasible bid profile $\boldsymbol{d} = (d_1, d_2, \ldots, d_n)$ of n bidders defined by Equation (11) is a Nash equilibrium.*

Before giving an outline of the proof of Theorem 9, we give simple examples.

Example 1. Let $N = \{1, 2\}$, $M = \{1, 2, 3\}$, and $(v_1(0) = v_2(0) = 0)$

$$v_1(1) = v_1(2) = 6, \quad v_1(3) = 12, \qquad v_2(1) = v_2(2) = 4, \quad v_2(3) = 8. \tag{12}$$

Then each v_i $(i \in N)$ satisfies Definition 2, and $(w_1(0) = w_2(0) = 0)$

$$w_1(1) = 6, \quad \tfrac{w_1(2)}{2} = 3, \quad \tfrac{w_1(3)}{3} = 3, \qquad w_2(1) = 4, \quad \tfrac{w_2(2)}{2} = 2, \quad \tfrac{w_2(3)}{3} = 2.$$

In this case, there is no Nash equilibrium. Actually, by symmetry, we can assume there are only four distinct partitions $M_1^{(k)}, M_2^{(k)} = M - M_1^{(k)}$ $(k = 0, 1, 2, 3)$ of M with $M_1^{(k)} = \{j \in M \mid j \le k\}$. Thus, $M_1^{(0)} = \emptyset$, $M_1^{(1)} = \{1\}$, $M_1^{(2)} = \{1, 2\}$, $M_1^{(3)} = \{1, 2, 3\}$. Corresponding to the partition $P^{(k)} = (M_1^{(k)}, M_2^{(k)})$ of M, the feasible bid profile $\boldsymbol{d}^{(k)} = (d_1^{(k)}, d_2^{(k)})$ defined by Equation (11) will be

$$\begin{array}{llll} d_1^{(0)} = (0,0,0), & d_1^{(1)} = (6,0,0), & d_1^{(2)} = (3,3,0), & d_1^{(3)} = (3,3,3), \\ d_2^{(0)} = (2,2,2), & d_2^{(1)} = (0,2,2), & d_2^{(2)} = (0,0,4), & d_2^{(3)} = (0,0,0). \end{array}$$

For $k = 0, 1, 2$, let bidder 1 change bid $d_1^{(k)}$ to $d_1'^{(k)}$ defined as follows:

$$d_1'^{(0)} = (6,0,0), \qquad d_1'^{(1)} = (1.6, 2.2, 2.2), \qquad d_1'^{(2)} = (0.8, 0.8, 4.4).$$

Then it is easy to see that bidder 1 can improve his payoff in the feasible bid profile $\boldsymbol{d}_1'^{(k)} = (d_1'^{(k)}, d_2^{(k)})$ for $k = 0, 1, 2$. Similarly, for $k = 3$, if bidder 2 changes bid $d_2^{(3)}$ to $d_2'^{(3)} = (0, 0, 4)$ then bidder 2 can improve her payoff in the feasible bid profile $\boldsymbol{d}_2'^{(3)} = (d_1^{(3)}, d_2'^{(3)})$.

By Theorem 9, the valuation profile $\boldsymbol{v} = (v_1, v_2)$ in Equation (12) has no Nash equilibrium. □

Example 2. Let $N = \{1, 2\}$, $M = \{1, 2, 3, 4, 5\}$ and $(v_i(0) = 0)$

$$v_i(1) = v_i(2) = v_i(3) = 3, \qquad v_i(4) = v_i(5) = 6 \tag{13}$$

for each $i \in N$. Then each v_i $(i \in N)$ satisfies Definition 2 and $(w_i(0) = 0)$

$$w_i(1) = 3, \quad \tfrac{w_i(2)}{2} = 1.5, \quad \tfrac{w_i(3)}{3} = \tfrac{w_i(4)}{4} = \tfrac{w_i(5)}{5} = 1.$$

As in Example 1, for $M_1^{(3)} = \{1, 2, 3\}$ and $M_2^{(3)} = \{4, 5\}$, $\boldsymbol{d}^{(3)} = (d_1^{(3)}, d_2^{(3)})$ in (11) is

$$d_1^{(3)} = (1, 1, 1, 0, 0), \qquad d_2^{(3)} = (0, 0, 0, 1.5, 1.5).$$

The feasible bid profile $\boldsymbol{d}^{(3)} = (d_1^{(3)}, d_2^{(3)})$ with $M_1^{(3)} = \{1, 2, 3\}$ and $M_2^{(3)} = \{4, 5\}$ is not a Nash equilibrium. However, $\boldsymbol{d}^{(1)} = (d_1^{(1)}, d_2^{(1)})$ with $d_1^{(1)} = (3, 0, 0, 0, 0)$, $d_2^{(1)} = (0, 1, 1, 1, 1)$ $(M_1^{(1)} = \{1\}$, $M_2^{(1)} = \{2, 3, 4, 5\})$ and $\boldsymbol{d}^{(4)} = (d_1^{(4)}, d_2^{(4)})$ with $d_1^{(4)} = (1, 1, 1, 1, 0)$, $d_2^{(4)} = (0, 0, 0, 0, 3)$ $(M_1^{(4)} = \{1, 2, 3, 4\}$, $M_2^{(4)} = \{5\})$ are both Nash equilibria. □

We give an outline of the proof of Theorem 9 using the following notation.

For a bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$, let $Y_i = X_i(\boldsymbol{b})$ be the set of items allocated to bidder i and let $y_i = |Y_i|$ $(i = 1, 2, \ldots, n)$. Then, clearly, $Y_i \cap Y_h = \emptyset$ $(i, h \in N, i \ne h)$ and $y_1 + y_2 + \cdots + y_n = m$. Thus, $P = (Y_1, Y_2, \ldots, Y_n)$ is a partition of M into n subsets, and if we let $M_i = Y_i$ then Equation (10) is satisfied. Furthermore, let $c_i = (c_i(1), c_i(2), \ldots, c_i(m))$ be the bid d_i of bidder i defined by Equation (11) in this case. Thus, we can write $c_i = (c_i(1), c_i(2), \ldots, c_i(m))$ as follows:

$$c_i(j) = \begin{cases} \frac{w_i(y_i)}{y_i} & (j \in Y_i), \\ 0 & (j \in M - Y_i). \end{cases} \tag{14}$$

Then we have the following lemma (we will give its proof in Section 5).

Lemma 10 *In a valuation profile* $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$*, if a feasible bid profile* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ *is a Nash equilibrium, then* $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ *defined by (14) is also a Nash equilibrium.*

Using this lemma, we can easily prove Theorem 9 as follows.

Proof of Theorem 9 If there is a feasible bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ which is a Nash equilibrium, then, by Lemma 10, $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ defined by Equation (14) is also a Nash equilibrium. Thus, by setting $M_i = Y_i$ and $d_i = c_i$ for each $i \in N$, we have a desired partition of M into n subsets and the necessity of Theorem 9 is proved.

Sufficiency is trivial. If there is a partition $P = (M_1, M_2, \ldots, M_n)$ of M into n subsets such that the feasible bid profile $\boldsymbol{d} = (d_1, d_2, \ldots, d_n)$ of n bidders defined by Equation (11) is a Nash equilibrium, then it is clearly a Nash equilibrium in the valuation profile $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$. □

4 Basic properties of a feasible bid profile $\boldsymbol{b}$

Before giving an outline of the proof of Lemma 10, we examine basic properties of a feasible bid profile. Here, we assume that $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is a feasible bid profile of n bidders and that, as mentioned before, all elements of each b_i $(i \in N)$ are ordered by using some permutation π_i on $M = \{1, 2, \ldots, m\}$ as follows:

$$b_i(\pi_i(1)) \leq b_i(\pi_i(2)) \leq \cdots \leq b_i(\pi_i(m)). \tag{15}$$

We also assume, all elements of each $b_{-i}^{\max} = (b_{-i}^{\max}(1), b_{-i}^{\max}(2), \ldots, b_{-i}^{\max}(m))$ $(i \in N)$ defined by Equation (1) are ordered by using a permutation π_{-i} on M as follows:

$$b_{-i}^{\max}(\pi_{-i}(1)) \leq b_{-i}^{\max}(\pi_{-i}(2)) \leq \cdots \leq b_{-i}^{\max}(\pi_{-i}(m)). \tag{16}$$

To prove Lemma 10, we need the notion of prestability and stability.

For a bid vector $b = (b(1), b(2), \ldots, b(m))$, let $b(j \leftrightarrow j')$ be the bid vector obtained from b by swapping $b(j)$ and $b(j')$. For example, if $b = (b(1), b(2), b(3))$ then $b(1 \leftrightarrow 3) = (b(3), b(2), b(1))$.

Definition 11 *Let* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ *be a feasible bid profile. For* $i \in N$*, let* $X_i(\boldsymbol{b})$ *be the set of items allocated to bidder* i*. Then* b_i *is called prestable in* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$*, if*

$$u_i(X_i(\boldsymbol{b})) \geq u_i(X_i(\boldsymbol{b}_i'))$$

for all feasible bid profiles $\boldsymbol{b}_i' = (b_i', \boldsymbol{b}_{-i})$ *with* $b_i' = b_i(j \leftrightarrow j')$ *(*$1 \leq j \neq j' \leq m$*) and* $|X_i(\boldsymbol{b}_i')| = |X_i(\boldsymbol{b})|$*. Otherwise,* b_i *is called unprestable in* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$*. If all* b_i $(i \in N)$ *are prestable in* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$*, then* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ *is calld prestable.*

Note that, by the definition, a prestable bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is always feasible and that, if some b_i is unprestable in a feasible bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$, then $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is not a Nash equilibrium. Furthermore, we have the following lemma (see Appendix for its proof).

Lemma 12 *For a prestable bid profile* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$*, let* $Y_i = X_i(\boldsymbol{b})$ *be the set of items allocated to bidder* $i \in N$ *and let* $y_i = |Y_i|$*. Then we can always choose a permutation* π_{-i} *on* $M = \{1, 2, \ldots, m\}$ *appropriately such that*

$$b_{-i}^{\max}(\pi_{-i}(1)) \leq b_{-i}^{\max}(\pi_{-i}(2)) \leq \cdots \leq b_{-i}^{\max}(\pi_{-i}(m)), \quad \textit{and} \tag{17}$$

$$Y_i = \{\pi_{-i}(1), \pi_{-i}(2), \ldots, \pi_{-i}(y_i)\}. \tag{18}$$

Definition 13 *Let* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ *be a prestable bid profile satisfying (17) and (18), where* $Y_i = X_i(\boldsymbol{b})$ *is the set of items allocated to bidder* $i \in N$ *and* $y_i = |Y_i|$ *(thus,* $P = (Y_1, Y_2, \ldots, Y_n)$ *is a partition of* M *into* n *subsets and* $y_1 + y_2 + \cdots + y_n = m$*). For* $i \in N$*, if*

$$v_i(y_i + k) - v_i(y_i) \leq \sum_{j=1}^{k} b_{-i}^{\max}(\pi_{-i}(y_i + j)) \tag{19}$$

for all k *with* $1 \leq k \leq m - y_i$ *and*

$$v_i(y_i - k') \leq v_i(y_i) - \sum_{j=0}^{k'-1} b_{-i}^{\max}(\pi_{-i}(y_i - j)) \tag{20}$$

for all k' *with* $1 \leq k' \leq y_i$*, then* b_i *is called stable in* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ *(otherwise it is called unstable). If all* b_i $(i \in N)$ *are stable in* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$*, then* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ *is calld stable.*

If a prestable bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is stable, then even if bidder i changes b_i to b_i' which is feasible or not, the payoff of bidder i will not increase in $(b_i', \boldsymbol{b}_{-i})$. Thus, any prestable bid profile $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ which is stable is a Nash equilibrium. The converse is also true and we have the following theorem (see Appendix for its proof).

Theorem 14 *Let* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ *be a prestable bid profile satisfying (17) and (18), where* $Y_i = X_i(\boldsymbol{b})$ *is the set of items allocated to bidder* $i \in N$ *and* $y_i = |Y_i|$*. Then* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ *is a Nash equilibrium if and only if* $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ *is stable.*

By Theorems 6 and 14, we can determine whether $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is a Nash equilibrium or not in $O(mn)$ time. Furthermore, by combining this with Theorem 9, we can determine whether there is a Nash equilibrium or not in a valuation profile $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ satisfying Definition 2 in $O(mn\binom{m+n-1}{n-1})$ time where $\binom{n}{k}$ is a binomial coefficient and, if there is, we can find such a Nash equilibrium in $O(mn\binom{m+n-1}{n-1})$ time.

From this theorem, we can also obtain the proof of Lemma 10 without much difficulty.

5 Proof of Lemma 10

Finally, we study properties of $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ defined by (14) and complete the proof of Lemma 10. For each $i \in N$, let $X_i(\boldsymbol{c})$ be the set of items allocated to bidder i in $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$. Thus,

$$X_i(\boldsymbol{c}) = Y_i, \quad |X_i(\boldsymbol{c})| = y_i \quad (i = 1, 2, \ldots, n), \quad \text{and} \quad y_1 + y_2 + \cdots + y_n = m. \tag{21}$$

We order the items not contained in $X_i(\boldsymbol{c})$ in nondecreasing order in $b_{-i}^{\max}$. Thus, we can assume that the items in $M - X_i(\boldsymbol{c}) = \{j_1^{(-i)}, j_2^{(-i)}, \ldots, j_{m-y_i}^{(-i)}\}$ are ordered as follows:

$$b_{-i}^{\max}(j_1^{(-i)}) \leq b_{-i}^{\max}(j_2^{(-i)}) \leq \cdots \leq b_{-i}^{\max}(j_{m-y_i}^{(-i)}). \tag{22}$$

Similarly, we consider $c_{-i}^{\max}$ where $c_{-i}^{\max}(j) = \max_{h \in N - \{i\}}\{c_h(j)\}$ for each $j \in M$ and order the items in $M - X_i(\boldsymbol{c}) = \{j_1^{(-i)}, j_2^{(-i)}, \ldots, j_{m-y_i}^{(-i)}\}$ in nondecreasing order in $c_{-i}^{\max}$ by using a permutation σ_{-i} as follows:

$$c_{-i}^{\max}(\sigma_{-i}(j_1^{(-i)})) \leq c_{-i}^{\max}(\sigma_{-i}(j_2^{(-i)})) \leq \cdots \leq c_{-i}^{\max}(\sigma_{-i}(j_{m-y_i}^{(-i)})). \tag{23}$$

Then the following lemma holds (see Appendix for its proof).

Lemma 15 *For* $i \in N$*, let* $k_i \leq m - y_i$ *be a nonnegative integer. Then*

$$\sum_{h=1}^{k_i} b_{-i}^{\max}(j_h^{(-i)}) \leq \sum_{h=1}^{k_i} c_{-i}^{\max}(\sigma_{-i}(j_h^{(-i)})), \tag{24}$$

i.e., the sum of the k_i *smallest bids for the items in* $M - X_i(\boldsymbol{c}) = \{j_1^{(-i)}, j_2^{(-i)}, \ldots, j_{m-y_i}^{(-i)}\}$ *in* $b_{-i}^{\max}$ *is at most the sum of the* k_i *smallest bids for the items in* $M - X_i(\boldsymbol{c})$ *in* $c_{-i}^{\max}$*.*

By using this lemma and Theorem 14, we can obtain the proof of Lemma 10.

Proof of Lemma 10: Suppose contrary that, $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ were not a Nash equilibrium even though $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is a Nash equilibrium. Then there is a bidder $i \in N$ such that if i changes the bid then in the resulting bid profile i will obtain more payoff. For simplicity, we assume $i = 1$ by symmetry. Thus, we can assume bidder 1 changes c_1 to c_1' and his payoff $u_1(X_1(c_1', \boldsymbol{c}_{-1}))$ of $X_1(c_1', \boldsymbol{c}_{-1})$ of items allocated to him in the bid profile $(c_1', \boldsymbol{c}_{-1})$ is greater than his payoff $u_1(X_1(\boldsymbol{c}_1))$ in the bid profile $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$. Thus, we have

$$u_1(X_1(c_1', \boldsymbol{c}_{-1})) = v_1(|X_1(c_1', \boldsymbol{c}_{-1})|) - \sum_{j \in X_1(c_1', \boldsymbol{c}_{-1})} c_{-1}^{\max}(j) > u_1(X_1(\boldsymbol{c})) = v_1(y_1). \tag{25}$$

We will show below that this leads to a contradiction.

We can assume $X_1(c_1', \boldsymbol{c}_{-1}) \supseteq X_1(\boldsymbol{c})$. Actually, by the definition of $\boldsymbol{c}$, we have $c_{-1}^{\max}(j) = 0$ for every $j \in X_1(\boldsymbol{c})$, and, by the monotonicity of v_1, we can modify $c_1'(j)$ so that $X_1(c_1', \boldsymbol{c}_{-1})$ may include j without decreasing the value of $u_1(X_1(c_1', \boldsymbol{c}_{-1}))$ (by deleting an item in $X_1(c_1', \boldsymbol{c}_{-1}) - X_1(\boldsymbol{c})$ if necessary). Now let $M_{-1} = X_1(c_1', \boldsymbol{c}_{-1}) - X_1(\boldsymbol{c})$ and $k_1 = |M_{-1}|$. Then we can write

$$u_1(X_1(c_1', \boldsymbol{c}_{-1})) = v_1(|X_1(c_1', \boldsymbol{c}_{-1})|) - \sum_{j \in M_{-1}} c_{-1}^{\max}(j). \tag{26}$$

Since $X_1(c_1', \boldsymbol{c}_{-1}) = X_1(\boldsymbol{c}) \cup M_{-1}$ and $|X_1(c_1', \boldsymbol{c}_{-1})| = |X_1(\boldsymbol{c}_1)| + |M_{-1}| = y_1 + k_1$, by applying Lemma 15, we have

$$\begin{aligned} u_1(X_1(c_1', \boldsymbol{c}_{-1})) &= v_1(|X_1(c_1', \boldsymbol{c}_{-1})|) - \sum_{j \in M_{-1}} c_{-1}^{\max}(j) \\ &\leq v_1(|X_1(c_1', \boldsymbol{c}_{-1})|) - \sum_{h=1}^{k_1} b_{-1}^{\max}(j_h^{(-i)}) = v_1(y_1 + k_1) - \sum_{h=1}^{k_1} b_{-1}^{\max}(j_h^{(-i)}) \end{aligned}$$

by Equation (26). Moreover, since $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is a Nash equilibrium, we have

$$v_1(y_1 + k_1) - v_1(y_1) \leq \sum_{h=1}^{k_1} b_{-1}^{\max}(j_h^{(-i)})$$

by Definition 13 and Theorem 14. By combining these, we have $u_1(X_1(c_1', \boldsymbol{c}_{-1})) \leq v_1(y_1)$, however, this contradicts $u_1(X_1(c_1', \boldsymbol{c}_{-1})) > u_1(X_1(\boldsymbol{c})) = v_1(y_1)$ in (25). Thus, $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ is a Nash equilibrium. □

6 Concluding remarks

We gave a necessary and sufficient condition for a valuation profile $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ satisfying Definition 2 to have a Nash equilibrium in Theorem 9. We give a remark that if all valuations v_i are submodular and symmetric then a Nash equilibrium that maximizes the social welfare (thus, it is optimal) can be obtained in polynomial time in n and m, however, the price of anarchy is still 2. Finally, we ask the following questions. Is there a polynomial time algorithm to decide whether the model in this paper has a Nash equilibrium or not? Is it possible to relax the constraint of symmetry in a valuation and to obtain a similar result which might lead to an answer to the open question in [1]?

Recently, Dobzinski, Fu, and Kleinberg published that it takes exponential communication to find a pure no-overbidding Nash equilibrium in combinatorial auctions with subadditive bidders, even if such equilibrium is known to exist [4]. However, this does not settle the open question posed by Bhawalkar and Roughgarden. Note also that, this does not imply that any algorithm for deciding whether there is a pure no-overbidding Nash equilibrium in combinatorial auctions with subadditive bidders requires exponential time.

References

[1] K. Bhawalkar and T. Roughgarden, Welfare guarantees for combinatorial auctions with item bidding, in: *Proc. of 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 700–709, 2011.

[2] A. Blume, P. Heidhues, J. Lafky, J. Münster, and M. Zhang, All Nash Equilibria of the Multi-Unit Vickrey Auction, in: *SFB/TR 15 Discussion Paper 116*, 2006.

[3] G. Chrisodoulou, A. Kovács, and M. Schapira, Bayesian combinatorial auctions, in: *Proc. of 35th ICALP*, pp.820–832, 2008.

[4] S. Dobzinski, H. Fu, and R. Kleinberg, On the complexty of computing an equilibrium in combinatorial auctions, in: *Proc. of 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 110–122, 2015.

[5] S. Dobzinski, N. Nisan, and M. Schapira, Approximation Algorithms for combinatorial auctions with complement-free bidders, in: *Proc. of 37th Annual ACM Symposium on Theory of Computing*, pp. 610–618, 2005.

[6] S. Dobzinski and M. Schapira, An improved approximation algorithm for combinatorial auctions with submodular bidders, in: *Proc. of 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1064–1073, 2006.

[7] Debt Management Report 2011, The Government Debt Management and the State of Public Debts, Financial Bureau, Ministry of Finance, Japan, pp. 34-46, http://www.mof.go.jp/english/jgbs/publication/debt_management_ report/2011/saimu.pdf.

[8] U. Feige, On maximizing welfare where utility functions are subadditive, in: *Proc. of 38th Annual ACM Symposium on Theory of Computing*, pp. 41–50, 2006 (see also *SIAM J. Computing*, 39, pp. 122–142, 2009).

[9] B. de Keijzer, E. Markakis, G. Schäfer, and O. Telelis, Inefficiency of standard multi-unit auctions, in: *Proc. of 21th Annual European Symposium on Algorithms*, pp. 385–396, 2013.

[10] S. Khot, R. Lipton, E.Markakis, and A. Mehta, Inapproximability results for combinatorial auctions with submodular utility functions, in: *Proc. of WINE 2005, Lecture Notes in Computer Science 3828* pp. 92–28, 2005.

[11] A. M. Kwasnica, K. Sherstyuk, Multi-unit auctions, in: *Journal of Economic Surveys*, 27.3, pp. 461–490, 2013.

[12] B. Lehmann, D. Lehmann, and N. Nisan, Combinatorial auctions with decreasing marginal utilities, in: *Proc. of 3rd Annual ACM Symposium on Electronic Commerce*, pp. 18–28, 2001.

[13] N. Nisan, Bidding and allocation in combinatorial auctions, in: *Proc. of 2nd Annual ACM Symposium on Electronic Commerce*, pp. 1–12, 2000.

[14] H. Umeda and T. Asano, Unpublished note, Department of Information and System Engineering, Chuo University, 2017.

[15] J. Vondrák, Optimal approximation for the submodular welfare problem in the value oracle model, in: *Proc. of 40th Annual ACM Symposium on Theory of Computing*, pp. 67–74, 2008.

Strengthening some complexity results on toughness of graphs

Kitti Varga[1]

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
vkitti@cs.bme.hu

Abstract: Let t be a positive real number. A graph is called t-tough, if the removal of any cutset S leaves at most $|S|/t$ components. The toughness of a graph is the largest t for which the graph is t-tough. In this paper we prove that for any positive rational number t, deciding whether $\tau(G) = t$ is DP-complete and if $t < 1$, this problem remains DP-complete for bipartite graphs. We also show that for any integer $r \geq 4$, if G is an r-regular bipartite graph, deciding whether $\tau(G) = 1$ is coNP-complete, and for any integer $k \geq 2$ and positive rational number $t \leq 1$, if G is a k-connected bipartite graph, deciding whether $\tau(G) \geq t$ is coNP-complete.

Keywords: toughness, complexity, bipartite graphs

1 Introduction

All graphs considered in this paper are finite, simple and undirected. Let $\omega(G)$ denote the number of components and $\alpha(G)$ denote the independence number.

Definition 1 *Let t be a positive real number. A graph G is called* t-tough, *if*

$$\omega(G - S) \leq \frac{|S|}{t}$$

for any cutset S of G. The toughness *of G, denoted by $\tau(G)$, is the largest t for which G is t-tough, taking $\tau(K_n) = \infty$ for all $n \geq 1$.*

We say that a cutset $S \subseteq V(G)$ is a tough set *if $\omega(G - S) = |S|/\tau(G)$.*

Let t be an arbitrary positive rational number and consider the following problem.

***t*-Tough**
Instance: a graph G.
Question: is it true that $\tau(G) \geq t$?

It is easy to see that for any $t \in \mathbb{Q}^+$, t-Tough is in coNP: a witness is a cutset S whose removal from the graph leaves more than $|S|/t$ components. Bauer et al. proved that this problem is coNP-complete.

Theorem 2 ([3]) *For any positive rational number t, t-Tough is coNP-complete.*

They also proved that t-Tough is coNP-complete for at least 3 regular graphs.

Theorem 3 ([2]) *For any fixed integer $r \geq 3$, 1-Tough is coNP-complete for r-regular graphs.*

[1]Research is supported by the National Research, Development and Innovation Office – NKFIH, No. 108947

Obviously, the toughness of a bipartite graph is at most one, since the removal of the smaller (or nonlarger) colorclass leaves the remaining vertices isolated. However, this fact does not make the problem 1-TOUGH easier for bipartite graphs.

Theorem 4 ([5]) 1-TOUGH *is coNP-complete for bipartite graphs.*

The complexity class DP was introduced by C. H. Papadimitriou and M. Yannakakis [6].

Definition 5 *A language L is in the class* DP *if there exist two languages $L_1 \in NP$ and $L_2 \in coNP$ such that $L = L_1 \cap L_2$.*

A language is called DP-hard if all problems in DP can be reduced to it. A language is DP-complete if it is in DP and it is DP-hard.

We mention that DP $\neq$ NP $\cap$ coNP, if NP $\neq$ coNP. Moreover, NP $\cup$ coNP $\subseteq$ DP. Many exact version of NP-complete (or coNP-complete) problems are DP-complete, and now we present some related ones.

EXACTCLIQUE
Instance: a graph G and a positive rational number k.
Question: is it true that the largest clique of G has size exactly k?

Theorem 6 ([6]) EXACTCLIQUE *is DP-complete.*

EXACTINDEPENDENCENUMBER
Instance: a graph G and a positive rational number k.
Question: is it true that $\alpha(G) = k$?

By taking the complement of the graph, we can obtain EXACTINDEPENDENCENUMBER from EXACTCLIQUE.

Corollary 7 EXACTINDEPENDENCENUMBER *is DP-complete.*

Let t be an arbitrary positive rational number and consider the following problem.

EXACT-t-TOUGH
Instance: a graph G.
Question: is it true that $\tau(G) = t$?

Our main result is that the exact version of the coNP-complete problem t-TOUGH is DP-complete, even for bipartite graphs.

Theorem 8 *For any positive rational number t,* EXACT-t-TOUGH *is DP-complete.*

Theorem 9 *For any positive rational number $t < 1$,* EXACT-t-TOUGH *remains DP-complete for bipartite graphs.*

Our constructions also give alternative proofs for Theorem 2 and Theorem 4. Furthermore, we also prove that 1-TOUGH remains coNP-complete for at least 4 regular bipartite graphs and for k-connected bipartite graphs, where $k \geq 2$. The cases of 1-tough 3-connected bipartite graphs and 3-regular bipartite graphs were asked in [1]. The second problem remains open.

Theorem 10 *For any fixed integer $r \geq 4$,* 1-TOUGH *remains coNP-complete for r-regular bipartite graphs.*

Theorem 11 *For any fixed integer $k \geq 2$ and positive rational number $t \leq 1$,* t-TOUGH *remains coNP-complete for k-connected bipartite graphs.*

This paper is structured as follows. After proving some useful lemmas, we prove Theorem 8 in Section 3. In Section 4 we prove the other three theorems about bipartite graphs, Theorem 9, 10 and 11.

2 Preliminaries

In this section we prove some useful lemmas.

Proposition 12 *Let G be a connected noncomplete graph on n vertices. Then $\tau(G) \in \mathbb{Q}^+$, and if $\tau(G) = a/b$, where a, b are positive integers and $(a, b) = 1$, then $1 \leq a, b \leq n - 1$.*

PROOF: By definition,

$$\tau(G) = \min_{\substack{S \subseteq V(G) \\ \text{cutset}}} \frac{|S|}{\omega(G - S)}$$

for a noncomplete graph G. Since G is connected and noncomplete, $1 \leq |S| \leq n - 2$ and since S is a cutset, $2 \leq \omega(G - S) \leq n - 1$. □

Corollary 13 *Let G and H be two connected noncomplete graphs on n vertices. If $\tau(G) \neq \tau(H)$, then*

$$|\tau(G) - \tau(H)| > \frac{1}{n^2}.$$

Claim 14 *For any positive rational number t,* EXACT-t-TOUGH $\in DP$.

PROOF: For any positive rational number t,

$$\text{EXACT-}t\text{-TOUGH} = \{G \text{ graph} \mid \tau(G) = t\} = \{G \text{ graph} \mid \tau(G) \geq t\} \cap \{G \text{ graph} \mid \tau(G) \leq t\}.$$

Let

$$L_1 = \{G \text{ graph} \mid \tau(G) \leq t\}$$

and

$$L_2 = \{G \text{ graph} \mid \tau(G) \geq t\}.$$

Then $L_2 \in$ coNP (a witness is a cutset $S \subseteq V(G)$ whose removal leaves more than $|S|/t$ components). Now we show that $L_1 \in$ NP, i.e. we can express L_1 in a form of

$$L_1 = \{G \text{ graph} \mid \tau(G) < t + \varepsilon\},$$

which is a complementer of a language belonging to coNP.

Let a, b be positive integers such that $t = a/b$ and $(a, b) = 1$, and let G be an arbitrary graph on n vertices. If G is disconnected, then $\tau(G) = 0$, and if G is complete, then $\tau(G) = \infty$, so in both cases $\tau(G) \neq t$. By Proposition 12, if $1 \leq a, b \leq n - 1$ does not hold, then again $\tau(G) \neq t$. So we can assume that $t = a/b$, where a, b are positive integers, $(a, b) = 1$ and $1 \leq a, b \leq n - 1$. With this assumption and by Corollary 13

$$L_1 = \{G \text{ graph} \mid \tau(G) \leq t\} = \left\{ G \text{ graph} \;\middle|\; \tau(G) < t + \frac{1}{|V(G)|^2} \right\},$$

so $L_1 \in$ NP, which means that EXACT-t-TOUGH $\in$ DP. □

Since the toughness of a bipartite graph can be at most 1, on the class of bipartite graphs EXACT-1-TOUGH and 1-TOUGH are the same, so we can conclude the following.

Corollary 15 *For any positive rational number $t < 1$,* EXACT-t-TOUGH-BIPARTITE *is in DP, and* EXACT-1-TOUGH-BIPARTITE *is in coNP.*

3 General graphs, proof of Theorem 8

In Claim 14 we have already proved that Exact-t-Tough $\in DP$. To prove Exact-t-Tough is DP-hard, we reduce ExactIndependenceNumber to it.

Proof: Let G be an arbitrary connected graph and let n denote the number of vertices in G and let $a, b \in \mathbb{N}$ be such that $t = a/b$. Let G_k be the following graph. For each $i \in [n]$ and $j \in [b]$, let

$$V_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,a}\}, \qquad V = \bigcup_{i=1}^{n} V_i,$$

$$U = \bigcup_{j=1}^{b} \bigcup_{i=1}^{n} u_{i,j}, \qquad U' = \{u'_1, \ldots, u'_{k(b-1)}\}, \qquad W = \{w_1, \ldots, w_{ak}\},$$

$$V(G_k) = V \cup U \cup U' \cup W.$$

For each $i \in [n]$ place a clique on V_i. For all $i, j \in [n]$, if $v_i v_j \in E(G)$, then place a complete bipartite graph on $(V_i; V_j)$. For each $i \in [n]$, $j \in [b]$ connect $u_{i,j}$ to every vertex of V_i. Connect every vertex in W to every vertex in $V \cup U \cup U'$. See Figure 1.

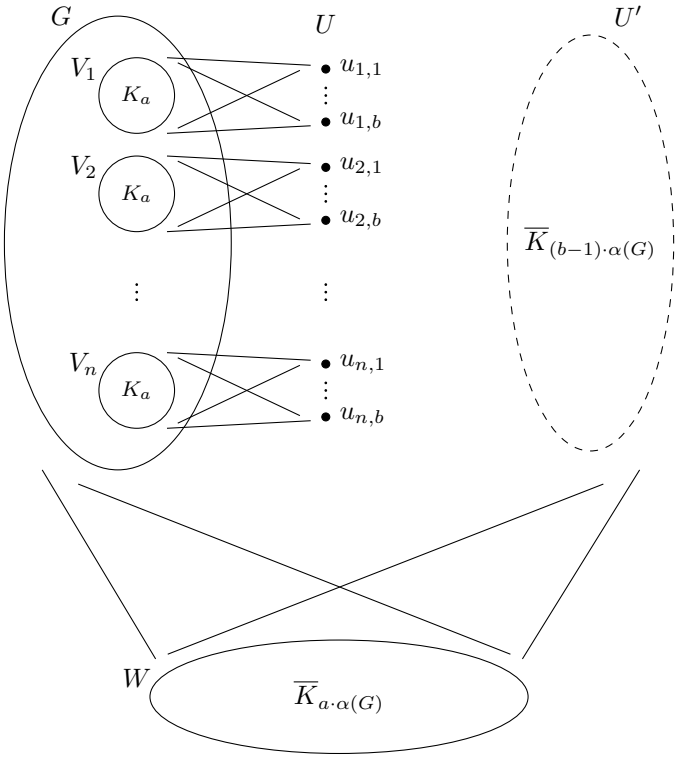


Figure 1: The graph G_k.

Obviously, G_k can be constructed from G in polynomial time. Now we show that $\alpha(G) = k$ if and only if $\tau(G_k) = t = a/b$.

Let $S \subseteq V(G_k)$ be an arbitrary cutset of G_k. Since S is a cutset, it must contain W. Let

$$I = \{i \in [n] \mid V_i \subseteq S\}.$$

After the removal of W, it does not "worth" it to remove any vertices of $U \cup U'$ or only a proper subset of V_i for any $i \in [n]$, so instead of S we can consider the vertex set

$$S' = S \setminus \left[(U \cup U') \cup \left(\bigcup_{i \notin I} V_i \right) \right].$$

Now

$$|S| \geq |S'| = |\{V_i \mid i \in I\}| + |W| = a|I| + ak$$

and

$$\omega(G_k - S) = \omega(G_k - S') \leq \alpha(G) + b|I| + k(b-1),$$

so

$$\frac{|S|}{\omega(G_k - S)} \geq \frac{|S'|}{\omega(G_k - S')} \geq \frac{a|I| + ak}{\alpha(G) + b|I| + k(b-1)}.$$

Now let $J \subseteq V(G)$ be an independent set of size $\alpha(G)$ in the graph G, and let

$$S_0 = \left(\bigcup_{i \notin J} V_i \right) \cup W.$$

Then

$$|S_0| = a\big(n - \alpha(G)\big) + ak$$

and

$$\omega(G_k - S_0) = \alpha(G) + b\big(n - \alpha(G)\big) + k(b-1),$$

so

$$\frac{|S_0|}{\omega(G_k - S_0)} = \frac{a\big(n - \alpha(G)\big) + ak}{\alpha(G) + b\big(n - \alpha(G)\big) + k(b-1)}.$$

Case 1: $\alpha(G) < k$. Then

$$\frac{|S|}{\omega(G_k - S)} \geq \frac{a|I| + ak}{\alpha(G) + b|I| + k(b-1)} > \frac{a(|I| + k)}{k + b|I| + k(b-1)} = \frac{a(|I| + k)}{b(|I| + k)} = \frac{a}{b} = t$$

for every cutset S of G_k, which implies that $\tau(G_k) > t$.

Case 2: $\alpha(G) = k$. Then

$$\frac{|S|}{\omega(G_k - S)} \geq \frac{a|I| + ak}{\alpha(G) + b|I| + k(b-1)} = \frac{a(|I| + k)}{k + b|I| + k(b-1)} = \frac{a(|I| + k)}{b(|I| + k)} = \frac{a}{b} = t$$

for every cutset S of G_k, which implies that $\tau(G_k) \geq t$.

Since

$$\tau(G_k) \leq \frac{|S_0|}{\omega(G_k - S_0)} = \frac{a\big(n - \alpha(G)\big) + ak}{\alpha(G) + b\big(n - \alpha(G)\big) + k(b-1)} = \frac{na}{nb} = \frac{a}{b} = t,$$

we can conclude that $\tau(G_k) = t$.

Case 3: $\alpha(G) > k$. Then

$$\tau(G_k) \leq \frac{|S_0|}{\omega(G_k - S_0)} = \frac{a\big(n - \alpha(G)\big) + ak}{\alpha(G) + b\big(n - \alpha(G)\big) + k(b-1)} <$$

$$< \frac{a\big(n - \alpha(G) + k\big)}{k + b\big(n - \alpha(G)\big) + k(b-1)} = \frac{a\big(n - \alpha(G) + k\big)}{b\big(n - \alpha(G) + k\big)} = \frac{a}{b} = t.$$

This means that $\alpha(G) = k$ if and only if $\tau(G_k) = t = a/b$. $\square$

The construction we used here is a slight modification of the one that Bauer et al. used in [4] for proving that recognizing $t \geq 1$ tough graphs is coNP-hard. They reduced a variant of IndependenceNumber to t-Tough.

Since in our proof $\alpha(G) \geq k$ if and only if $\tau(G_k) \geq t$, we can reduce IndependenceNumber to t-Tough, so this gives another proof for Theorem 2.

4 Bipartite graphs, proofs of Theorems 9, 10 and 11

Let G be an arbitrary connected graph on the vertices $v_1, \ldots, v_n$ and let $B(G)$ be the following bipartite graph. Let

$$V\big(B(G)\big) = \{v_{i,1}, v_{i,2} \mid i \in [n]\}$$

and for all $i, j \in [n]$, if $v_i v_j \in E(G)$, then connect $v_{i,1}$ to $v_{j,2}$ and $v_{i,2}$ to $v_{j,1}$. Also for all $i \in [n]$, connect $v_{i,1}$ to $v_{i,2}$, see Figure 2.

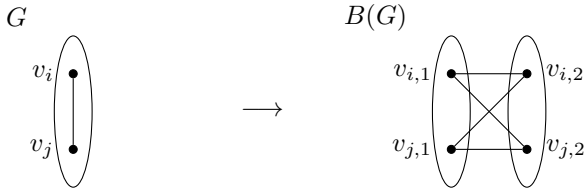


Figure 2: The construction of the graph $B(G)$.

Lemma 16 *Let G be an arbitrary connected graph with $\tau(G) \leq 1/2$. Then $\tau\big(B(G)\big) = 2\tau(G)$.*

Proof: Let $t = \tau(G)$, $G' = B(G)$ and let $S \subseteq V(G)$ be an arbitrary cutset in G. Consider the vertex set

$$S' = \{v_{i,1}, v_{i,2} \mid v_i \in S\}.$$

Clearly, S' is a cutset in G' and

$$\omega(G' - S') = \omega(G - S) \leq \frac{|S|}{t} = \frac{|S'|}{2t},$$

so $\tau(G') \leq 2t$.

Now let S' be an arbitrary cutset in G', and let

$$S'_1 = \{v_{i,1} \in S \mid v_{i,2} \notin S\} \cup \{v_{i,2} \in S \mid v_{i,1} \notin S\}$$

and

$$S'_2 = S' \setminus S'_1.$$

Consider those components of $G' - S'$, in which every vertex has its pair. Obviously S'_1 has "no effect" on these components, so (similarly as before) the number of these components is at most

$$\frac{|S'_2|}{2t}.$$

The number of the remaining components – so in which there is a vertex without its pair – can be at most $|S'_1|$, because the pair of the vertex mentioned before must be in S'_1. Since $t \leq 1/2$,

$$\omega(G' - S') \leq \frac{|S'_2|}{2t} + |S'_1| \leq \frac{|S'_2|}{2t} + \frac{|S'_1|}{2t} = \frac{|S'|}{2t}.$$

$\square$

Claim 17 *Let G be an arbitrary graph on the vertices $v_1, \ldots, v_n$ with $\tau(G) = t$. For all $i \in [n]$ add the vertex u_i to the graph and connect it to v_i and let H denote the obtained graph. Then*

$$\tau(H) = \min\left(\frac{t}{t+1}, \frac{1}{2}\right) = \begin{cases} \frac{t}{t+1}, & \text{if } t \leq 1, \\ \frac{1}{2}, & \text{if } t \geq 1, \end{cases}$$

and

$$\tau\big(B(H)\big) = \begin{cases} \frac{2t}{t+1}, & \text{if } t \leq 1, \\ 1, & \text{if } t \geq 1. \end{cases}$$

PROOF: Let S be an arbitrary cutset in H. Since the removal of a vertex of degree 1 does not disconnect anything in the graph, we can assume that $u_i \notin S$ for all $i \in [n]$, i.e. $S \subseteq V(G)$.

Case 1: S is a cutset in G. Since $\tau(G) = t$,

$$\omega(H - S) = \omega(G - S) + |S| \leq \frac{|S|}{t} + |S| = |S|\left(1 + \frac{1}{t}\right) = |S|\frac{t+1}{t}.$$

Moreover, if S is a tough set in G, then

$$\omega(H - S) = |S|\frac{t+1}{t}.$$

Case 2: S is not a cutset in G. Since S is a cutset in H, $S \neq \emptyset$, so

$$\omega(H - S) = 1 + |S| \leq 2|S|.$$

Moreover, if $|S| = 1$, then

$$\omega(H - S) = 2 = 2|S|.$$

This means that

$$\tau(H) = \min\left(\frac{t}{t+1}, \frac{1}{2}\right),$$

and by Lemma 16,

$$\tau\big(B(H)\big) = 2\tau(H).$$

□

Claim 18 *Let G be an arbitrary connected graph. Then $\tau\big(B(G)\big) = \min(2\tau(G), 1)$.*

PROOF: Let G be an arbitrary graph on the vertices $v_1, \ldots, v_n$ with $\tau(G) = t$. For all $i \in [n]$ add the vertex u_i to the graph and connect it to v_i and let H denote the obtained graph. Consider the graph $B(H)$ and the vertices $u_{1,1}$ and $u_{1,2}$. Since their open neighborhood, i.e. the set $\{v_{1,1}, v_{1,2}\}$ is connected,

$$\tau\big(B(H)\big) \leq \tau\big(B(H) - \{u_{1,1}, u_{1,2}\}\big).$$

Obviously, $B(G) = B(H) - \{u_{i,1}, u_{i,2} \mid i \in [n]\}$, so by Lemma 16 we can conclude that

$$\tau\big(B(G)\big) = \min(2\tau(G), 1).$$

□

Theorem 19 *For every positive rational number $t < 1$,* EXACT-t-TOUGH *remains DP-complete for bipartite graphs, and* EXACT-1-TOUGH *is coNP-complete for bipartite graphs.*

PROOF: In Corollary 15 we already showed that EXACT-t-TOUGH-BIPARTITE $\in DP$ if $t < 1$, and EXACT-1-TOUGH-BIPARTITE $\in coNP$.

We reduce EXACT-t-TOUGH to this problem.

Let G be an arbitrary connected graph. By Claim 18, for any positive rational number $t < 1/2$, $\tau(G) = t$ if and only if $\tau\big(B(G)\big) = 2t$, and for any positive rational number $t \geq 1/2$, $\tau(G) \geq t$ if and only if $\tau\big(B(G)\big) = 1$. $\square$

Corollary 20 *For any positive rational number $t \leq 1$, t-TOUGH remains coNP-complete for bipartite graphs.*

The case $t = 1$ was already proved by Kratsch et al. in [5]. In their proof the vertices $v_{i,1}$ and $v_{i,2}$ are not connected by an edge, but by a path with two inner vertices. With that construction the original graph is at least 1-tough if and only if the obtained bipartite graph is exactly 1-tough. However, due to the inner vertices of the pathes mentioned before, the constructed bipartite graph has many vertices of degree 2, so these graphs cannot be either regular (except cycles) or 3-connected.

Since in our proof if G is an r-regular graph, then $B(G)$ is an $(r+1)$-regular graph, so by Theorem 3 we can conclude the following.

Corollary 21 *For any fixed integer $r \geq 4$, 1-TOUGH remains coNP-complete for r-regular bipartite graphs.*

Moreover, in the proof if G is k-connected, then $B(G)$ is also k-connected.

Corollary 22 *For any fixed integer $k \geq 2$ and any positive rational number $t \leq 1$, t-TOUGH remains coNP-complete for k-connected bipartite graphs.*

References

[1] D. BAUER, J. VAN DEN HEUVEL, A. MORGANA, E. SCHMEICHEL, The complexity of recognizing tough cubic graphs, *Discrete Applied Mathematics* **79** (1997)

[2] D. BAUER, J. VAN DEN HEUVEL, A. MORGANA, E. SCHMEICHEL, The complexity of toughness in regular graphs, *Congressus Numerantium* **130** (1998)

[3] D. BAUER, S.L. HAKIM, E. SCHMEICHEL, Recognizing tough graphs is NP-hard, *Discrete Applied Mathematics* **28** (1990)

[4] D. BAUER, A. MORGANA, E. SCHMEICHEL, On the complexity of recognizing tough graphs, *Discrete Mathematics* **124** (1994)

[5] D. KRATSCH, J. LEHEL, H. MÜLLER, Toughness, hamiltonicity and split graphs, *Discrete Mathematics* **150** (1996)

[6] C. H. PAPADIMITRIOU, M. YANNAKAKIS, The complexity of facets (and some facets of complexity), *Journal of Computer and System Sciences* **28** (1984)

Spanning trees with few leaves in claw-free graphs

Gábor Wiener[1]

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics, Hungary
wiener@cs.bme.hu

Abstract: We generalize a theorem of Kano, Kyaw, Matsuda, Ozeki, Saito, and Yamashita [3] concerning the minimum number of leaves of spanning trees in claw-free graphs. The result also implies a strengthening of a result of Ainouche, Broersma, and Veldman [1].

Keywords: claw-free graph, spanning tree, leaf, DFS

1 Preliminaries

All graphs in this paper are simple, finite, and undirected; the vertex and edge sets of a graph G are denoted by $V(G)$ and $E(G)$, respectively. The degree of a vertex v of G is denoted by $deg_G(v)$ and the set of neighbours of v is denoted by $N_G(v)$. A graph is *claw-free* if it does not contain $K_{1,3}$ as an induced subgraph. A graph G is *traceable* if it contains a hamiltonian path. The *minimum leaf number* $\mathrm{ml}(G)$ is the minimum number of leaves (vertices of degree 1) of the spanning trees of G. The *minimum branch number* $\mathrm{s}(G)$ is the minimum number of branches (vertices of degree at least 3) of the spanning trees of G. A tree T is a *k-tree* if all vertices have degree at most k. The minimum sum of degrees of k independent vertices of G is denoted by $\delta_k(G)$.

Depth first search (DFS) (see for example [4]) is a traversal of a graph G; it visits the vertices of G one by one, such that an unvisited neighbour of the current vertex v is visited next provided there exists such a neighbour of v. If there is no unvisited neighbour of v then the algorithm steps back to the vertex u from which v was reached and continues the process from u as the current vertex. A unique *DFS number* is assigned to each vertex v, which is the rank of v in the order of visiting. Let v be a vertex of G. The vertex from which v was reached is called the *parent* of v and is denoted by $p(v)$, v is called a *child* of $p(v)$. It is obvious that the edges between vertices and their parents form a spanning tree T of G, a so-called *DFS-tree rooted* at the vertex r with DFS number 1. The vertices of the unique path between r and v are called the *ancestors* of v. A vertex having no child is called a *d-leaf* of T. Note that each d-leaf of T is also a leaf of T, and the root r is the only vertex that can be a leaf of T without being a d-leaf. A pair of vertices (a, b) is called a *cross edge* if none of a and b is an ancestor of the other. It is easy to see that an edge $(a, b) \in E(G)$ can not be a cross edge.

We conclude this section with a characterization of DFS-trees of a graph that will be useful later. For this, we need some further notions. Let T be a tree, $v \in V(T)$. A *rooting* of T at v is the directed graph obtained from T by directing the edges of T such that the unique path between v and an arbitrary vertex w is directed towards w. A rooting of a spanning tree T of a graph G is *nice*, if for every edge $(a, b) \in E(G)$ there is a directed path in the rooting from a to b or from b to a.

Lemma 1 *A spanning tree T of a graph G is a DFS-tree of G if and only if T has a nice rooting.*

[1]This research was supported by the National Research, Development and Innovation Office – NKFIH, grant no. OTKA 108947 and by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

PROOF: The "only if" part is straightforward, since if T is a DFS-tree of G, then there are no cross edges in G. Let us assume now that T has a nice rooting T' at $r \in V(T)$ and let us execute a depth first search starting at r, such that we prefer exploring the unvisited vertices of G through a directed edge of T' (that is, if we are at a vertex a and there exists a directed edge $(a, b) \in E(T')$ such that b is not yet visited, then we choose b as the next vertex). We claim that the DFS-tree obtained this way will be T. To prove this, assume to the contrary that there is a vertex where we choose to go through an edge not in T and let a have the smallest depth number among these vertices. Then there is a directed path P in T' from r to a, such that all vertices of P have been already visited. Let the edge we choose at a be $(a, b) \notin E(T)$. Since the rooting is nice, there is a directed path from a to b or from b to a in T'. In the first case we had to go through all the edges of the path before choosing the edge (a, b) and then b is already visited. In the second case b must be in P, since all vertices in T' has in-degree at most 1, which means that b is already visited, finishing the proof of the lemma. □

2 Introduction

Hamiltonian properties of claw-free graphs have been examined for more than three decades; one of the early results is due to Matthews and Sumner [6] and was also found independently by Liu, Tian, and Wu [5].

Theorem 2 (Matthews and Sumner, Liu et al., 1985) *Let G be a connected claw-free graph of order n. If $\delta_3(G) \geq n - 2$, then G is traceable.*

Gargano, Hammar, Hell, Stacho, and Vaccaro [2] proved a generalization of Theorem 2 concerning the minimum branch number.

Theorem 3 (Gargano et al., 2002) *Let G be a connected claw-free graph of order n and k a nonnegative integer. If $\delta_{k+3}(G) \geq n - k - 2$, then $\mathrm{s}(G) \leq k$.*

This result was generalized further by Salamon [8].

Theorem 4 (Salamon, 2010) *Let G be a connected claw-free graph of order n and $k \geq 2$ an integer. If $\delta_{k+1}(G) \geq n - k$, then $\mathrm{ml}(G) \leq k$.*

Since a branch vertex has degree at least 3, it is obvious that $\mathrm{ml}(G) \geq \mathrm{s}(G) + 2$, thus Theorem 4 is a generalization of Theorem 3 indeed. Theorem 4 was rediscovered in 2012 by Kano, Kyaw, Matsuda, Ozeki, Saito, and Yamashita [3] and they also proved a stronger version.

Theorem 5 (Kano et al., 2012) *Let G be a connected claw-free graph of order n and $k \geq 2$ an integer. If $\delta_{k+1}(G) \geq n - k$, then G has a spanning 3-tree with at most k leaves.*

The main result of the paper is the following theorem.

Theorem 6 *Every connected claw-free graph G has a DFS-tree T such that no two of the d-leaves of T have a common neighbour. Moreover, if v is not a cut vertex of G, then T can be chosen such that it is rooted at v.*

We prove Theorem 6 in the next section, here we show how Theorem 6 implies Theorem 5. Let G be a connected claw-free graph of order n with $\delta_{k+1}(G) \geq n - k$ and let T be a DFS-tree rooted at r, guaranteed by Theorem 6. The set of d-leaves D of T is obviously an independent set, since an edge between any two d-leaves would be a cross edge, thus the degree sum of the vertices of D is at most $n - |D|$, since all vertices in $V(G) - D$ has at most one neighbour in D. Hence $|D| \leq k$, that is T has at most $k + 1$ leaves.

Notice that T, like any DFS-tree of a claw-free graph is a 3-tree. In order to find a spanning 3-tree with at most k leaves, we need a further local improvement step. If T has at most $k-1$ d-leaves or $\deg_T(r) \neq 1$, then T has at most k leaves, thus we may suppose this is not the case. We may also assume that T is a minimum leaf spanning 3-tree of G and therefore there is no edge e between r and any of the d-leaves, otherwise by deleting an edge f incident with a degree 3 vertex of T from the unique cycle of $T+e$, we would obtain a spanning 3-tree with at most k leaves. (Such an edge f always exists, since T is not a hamiltonian path.) This means that the $k+1$ leaves of T form an independent set and thus their degree sum is at least $n-k$.

Let $B := \{\mathrm{p}(v) : v \in N_G(r)\}$ and let t be a degree 3 vertex of T. Then $t \notin B$: t has 2 children u and v and if any of them (say u) is in $N_G(r)$, then $T-(t,u)+(r,u)$ would be a spanning 3-tree with at most k leaves. This also shows that $t \notin N_G(r)$, otherwise $G[\{r,t,u,v\}]$ would be a claw. Let us observe that $|B| = |N_G(r)| = \deg_G(r)$, since no two vertices in $N_G(r)$ have the same parent.

Let C be the set of those vertices from $V(G)-D$ that are not neighbours of any of the d-leaves. For any $v \in N_G(r)$ we have $\mathrm{p}(v) \in B$, thus $\mathrm{p}(v)$ is either the root or has degree 2 in T. We claim that $\mathrm{p}(v) \in C$. Otherwise there is a d-leaf ℓ, such that $(\mathrm{p}(v),\ell) \in E(G)$, but $(\mathrm{p}(v),\ell) \notin E(T)$ (since $\deg_T(\mathrm{p}(v)) \neq 3$ and v is not a d-leaf). Now by deleting an edge f incident with a degree 3 vertex of T from the unique cycle of $T+(r,v)+(\mathrm{p}(v),\ell)-(v,\mathrm{p}(v))$, we would obtain a spanning 3-tree with at most k leaves (such an edge f exists again). That is, we have proved $B \subseteq C$. On the other hand, $|B| = \deg_G(r) \geq n-k-\sum_{\ell \in D}\deg_G(\ell) \geq n-k-(n-k-|C|) = |C|$, since any vertex of $V(G)-D-C$ is a neighbour of at most 1 d-leaf, and now we have $B=C$.

Let now t be a degree 3 vertex of T of minimum depth number (such a t exists, otherwise T is a hamiltonian path). We have seen that $t \notin N_G(r)$ and by the choice of t, $\deg_T(\mathrm{p}(t)) = 2$, that is $\mathrm{p}(t) \notin B = C$, thus there exists a d-leaf ℓ, such that $(\mathrm{p}(t),\ell) \in E(G)$. It is clear that $(\mathrm{p}(t),\ell) \notin E(T)$, thus $T-(\mathrm{p}(t),t)+(\mathrm{p}(t),\ell)$ is a spanning 3-tree of G with at most k leaves.

3 Proof of the main theorem

In this section we give a short proof of Theorem 6. The shortness of the proof is of some interest, since the proofs of Theorem 4 and 5 are quite lengthy. One of the reasons is that it is much easier to handle the cases of local improvements in DFS-trees than in general spanning trees. This shows that DFS can be a useful tool in similar problems as well.

Proof of Theorem 6: Let T be a DFS-tree of the claw-free graph G with a minimum number of d-leaves and among these with a minimum length sum of d-leaves, where the length of a d-leaf ℓ is defined as the length of the path between ℓ and the closest branch to ℓ in T. We claim that T has no d-leaves with a common neighbour. Assume to the contrary that there are d-leaves a and b, such that they have a common neighbour x. Then x is a common ancestor of a and b, since (x,a) and (x,b) are not cross edges. Now we distinguish three cases.

Case 1. $x \neq r$. Let x' be the parent of x and a' the parent of a in the DFS-tree T and let us consider $H = G[\{x,x',a,b\}]$. Since H is not a claw and $(a,b) \notin E(G)$, either $(x',a) \in E(G)$ or $(x',b) \in E(G)$. Suppose w. l. o. g. that $(x',a) \in E(G)$. Now it is easy to check using Lemma 1 that $T' = T-(x,x')-(a,a')+(x',a)+(a,x)$ is a DFS-tree of G with either a smaller number of d-leaves as T or with the same number of d-leaves, but a smaller length sum of the leaves as T (since x has two leaf descendants, there must be a branch on the path between x and a), contradicting the choice of T.

Case 2/a. $x = r$ and r has 1 child. Let x' be the child of x in the DFS-tree T and now the argument is the same as in Case 1.

Case 2/b. $x = r$ and r has at least 2 children. Let a' be the parent of a. Using Lemma 1 it is easy to see $T' = T-(a',a)+(a,r)$ is a DFS-tree of G with either a smaller number of d-leaves as T or with the same number of d-leaves, but a smaller length sum of the leaves as T, contradicting the choice of T.

To finish the proof of the theorem we have to show that if v is not a cut vertex of G, than the DFS-tree T can be chosen such that it is rooted at v. Let us observe that a DFS-tree with the extremal properties

required can be obtained from an arbitrary DFS-tree by executing the local improvement steps described above, while it is possible. Let us also observe that the root changes only when we execute the local improvement step corresponding to Case 2/b. That is, if the original DFS-tree has root v, then v remains the root till the end of the process, provided we do not have Case 2/b, which is obvious, since a root of a DFS-tree of degree at least 2 is a cut vertex of G. □

4 Other connections

Theorem 6 has another connection with results concerning claw-free graphs.

Corollary 7 *Let G be a connected claw-free graph of diameter at most 2 and let v be a non-cut vertex of G. Then there exists a hamiltonian path of G starting at v.*

Proof: By Theorem 6, there exists a DFS-tree T of G rooted at v, such that no two of the d-leaves of T have a common neighbour. Since the diameter of G is at most 2, this is possible only if T has just one d-leaf, which finishes the proof. □

Corollary 7 is a stronger form of a result of Ainouche, Broersma, and Veldman [1] stating that every connected claw-free graph of diameter at most 2 is traceable. Actually they also proved a much more general theorem. For a graph G, let G^2 denote the graph with the same vertices as G, such that there is an edge in G^2 between two vertices a and b if and only if their distance in G is at most 2.

Theorem 8 (Ainouche et al., 1990) *If G is an m-connected claw-free graph with $\alpha(G^2) \leq m+1$, then G is traceable.*

For graphs with higher connectivity Kano et al. in [3] conjectured an extension of Theorem 5.

Conjecture 9 (Kano et al., 2012) *Let G be an m-connected claw-free graph of order n and $k \geq m+1$ an integer. If $\delta_{k+1}(G) \geq n-k$, then* $\mathrm{ml}(G) \leq k-m+1$.

Remark 10 *Kano et al. mention that the conjecture is interesting only for $m \leq 6$, since Ryjáček proved [7] that every 7-connected claw-free graph is hamiltonian.*

Theorem 6 and Theorem 8 are both generalizations of the theorem concerning the traceability of connected claw-free graph of diameter at most 2, but in different directions. We make the following conjecture which is a common generalization of Theorem 8 and Conjecture 9.

Conjecture 11 *Let G be an m-connected claw-free graph of order n and $k \geq 2$ an integer. If $\alpha(G^2) \leq m+k-1$, then* $\mathrm{ml}(G) \leq k$.

It is easy to see that our conjecture is a common generalization of Theorem 8 and Conjecture 9, indeed: Theorem 8 is the special case $k=2$ of Conjecture 11 and $\delta_{k+1}(G) \geq n-k$ implies $\alpha(G^2) \leq k$, from which $\mathrm{ml}(G) \leq k-m+1$ follows, provided Conjecture 11 is true.

References

[1] A. Ainouche, H.J. Broersma, and H.J. Veldman, Remarks on hamiltonian properties of claw-free graphs, *Ars Combin.* **29C**, 110–121. (1990)

[2] L. Gargano, M. Hammar, P. Hell, L. Stacho, and U. Vaccaro, Spanning spiders and light-splitting switches, *Discrete Mathematics* **285**, 83–95. (2004)

[3] M. Kano, A. Kyaw, H. Matsuda, K. Ozeki, A. Saito, and T. Yamashita, Spanning trees with small number of leaves in a claw-free graph, *Ars Combin.* **103**, 137–154. (2012)

[4] J. van Leeuwen (ed.), Handbook of Theoretical Computer Science A: Algorithms and Complexity, Elsevier, Amsterdam (1990)

[5] Y. Liu, F. Tian, and Z. Wu, Some results on longest paths and cycles in $K_{1,3}$–free graphs, *J. Changsha Railway Inst.* **4**, 105–106. (1986)

[6] M.M. Matthews and D.P. Sumner, Longest paths and cycles in $K_{1,3}$–free graphs, *J. Graph Theory* **9** , 269–277. (1985)

[7] Z. Ryjáček, On a closure concept in claw-free graphs, *J. Combin. Theory Ser. B* **70**, 217–224. (1997)

[8] G. Salamon, Degree-Based Spanning Tree Optimization, PhD Thesis, Budapest University of Technology and Economics, `http://doktori.math.bme.hu/Ertekezesek/salamon_dissertation.pdf` (2010)

[9] C. Q. Zhang, Hamilton cycles in claw-free graphs, *J. Graph Theory* **12**, 209–216. (1988)

Author Index