"Almost stable" matchings in the Roommates problem with bounded preference lists

Péter Biró[1,*,†,‡], David F. Manlove[2,‡] and Eric J. McDermid[3,†,‡,§,¶]

[1] *Institute of Economics, Hungarian Academy of Sciences, Budapest, Hungary.*
Email: `birop@econ.core.hu`.

[2] *School of Computing Science, University of Glasgow, Glasgow, UK.*
Email: `david.manlove@glasgow.ac.uk`.

[3] *21st Century Technologies, Inc., 6011 W. Courtyard Drive, Austin, TX 78730.*
Email: `emcdermid@21technologies.com`.

Abstract

An instance of the classical Stable Roommates problem need not admit a stable matching. Previous work has considered the problem of finding a matching that is "as stable as possible", i.e., admits the minimum number of blocking pairs. It is known that this problem is NP-hard and not approximable within $n^{\frac{1}{2}-\varepsilon}$, for any $\varepsilon > 0$, unless P=NP, where n is the number of agents in a given instance. In this paper we extend the study to the Stable Roommates problem with Incomplete lists. In particular, we consider the case that the lengths of the lists are bounded by some integer d. We show that, even if $d = 3$, there is some $c > 1$ such that the problem of finding a matching with the minimum number of blocking pairs is not approximable within c unless P=NP. On the other hand we show that the problem is solvable in polynomial time for $d \leq 2$, and we give a $(2d-3)$-approximation algorithm for fixed $d \geq 3$. If the given lists satisfy an additional condition (namely the absence of a so-called *elitist odd party* – a structure that is unlikely to exist in general), the performance guarantee improves to $2d-4$.

1 Introduction

Background. The Stable Roommates problem (SR) has been the subject of much attention in the literature [6, 10, 7, 15, 16, 13, 1]. An instance of this problem comprises a set of n agents (where n is even), each of whom ranks all others in strict order of preference. A solution is a *stable matching*, which is a partition of the agents into pairs such that there is no *blocking pair* – this is a pair of agents, each of whom prefers the other to their partner in the matching. The Stable Roommates problem with Incomplete lists (SRI) is the generalisation of SR that arises when n need not be even, and agents can declare a subset of the others as being *unacceptable* (i.e., they can neither be matched to such

*Supported by OTKA grant K69027 and by the Hungarian Academy of Sciences under its Momentum Programme (LD-004/2010).

†Part of this work was carried out whilst at the School of Computing Science, University of Glasgow.

‡Supported by EPSRC grant EP/E011993/1.

§Part of this work was carried out whilst at the Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee.

¶Supported by NSF award CCF-0830678 and by the UWM Research Growth Initiative.

agents, nor form a blocking pair with them). A special bipartite restriction of SRI is the Stable Marriage problem with Incomplete lists (SMI), in which the agents are partitioned into two sets, the *men* and *women*, where men find only women acceptable and vice versa.

The full SR and SRI problem names reflect their original application to the assignment of students to share campus accommodation comprising two-person rooms, where students may have preferences over one another based on their compatibility as a roommate [6, 12]. However, a more recent application of SRI lies in kidney exchange [14], with each agent a_i corresponding to a pair (d_i, p_i), where d_i is a willing but incompatible donor for a patient p_i who requires a kidney transplant. Two agents $\{a_i, a_j\}$ find each other acceptable if d_i is compatible for p_j, and d_j is compatible for p_i (so that a *pairwise kidney exchange* is possible, in which d_i donates a kidney to p_j in exchange for d_j donating a kidney to p_i). Preference lists can reflect degrees of compatibility of patients for the donors from whom they can potentially receive a kidney. A stable matching in this context is one possible solution concept [14].

Gale and Shapley [6] showed that an SR instance I need not admit a stable matching. Irving [10] gave an $O(n^2)$ algorithm to find a stable matching in I or report that none exists. This algorithm has a straightforward $O(m)$ analogue for the case that I is an SRI instance [7, Section 4.5.2], where m is the number of mutually acceptable pairs of agents. By contrast, every instance of SMI admits a stable matching, which can be found in linear time by using the classical Gale-Shapley algorithm (see [6] and [7, Section 1.4.2]).

Related work. Since a stable matching need not exist in a given SRI instance I, various possibilities for coping with this situation have been formulated in the literature. Tan [16] gave an $O(m)$ algorithm for finding the smallest set of agents that need to be deleted from I in order to leave an instance with a stable matching. An alternative [1, 3] is to retain all agents in I and seek an "almost stable" matching, i.e., a matching M in I such that $|bp(M)|$ is minimum, where $bp(M)$ denotes the set of blocking pairs of M in I. The motivation for this problem is that, in many situations, agents' preferences are private, and there may be limited channels of communication that would lead to the awareness of blocking pairs in practice. Hence, whilst a blocking pair may exist in theory, in practice it may not lead to the matching in question being undermined. Naturally, the fewer the blocking pairs, the more likely this is to be true. Finding matchings with few blocking pairs has also been considered by a number of other authors in the context of SRI and SMI [11, 4, 8, 9, 5].

Abraham et al. [1] showed that the problem of finding a matching with the minimum number of blocking pairs in a given SR instance is NP-hard and not approximable within $n^{\frac{1}{2}-\varepsilon}$, for any $\varepsilon > 0$, unless P=NP. They improved this lower bound to $n^{1-\varepsilon}$ in the case that preference lists are permitted to contain ties.

"Almost stable" matchings have also been considered in the context of SMI. Whilst we have already noted that every SMI instance admits a stable matching [6], such a matching may be half the size of a maximum cardinality matching [3]. In applications where we seek to match as many agents as possible, a limited number of blocking pairs may be tolerated in order to arrive at a larger matching. In the SMI context, Biró et al. [3] proved that the problem of finding a maximum cardinality matching that admits the minimum number of blocking pairs is NP-hard and not approximable within $n^{1-\varepsilon}$, for any $\varepsilon > 0$, unless P=NP. Further, even if all preference lists are of length at most 3, they showed that the problem remains NP-hard and not approximable within c, for some constant $c > 1$. Hamada et al. [8] strengthened the latter result by improving the constant c to $n^{1-\varepsilon}$, for any $\varepsilon > 0$. For preference lists of length at most 2 on one side, Biró et al. [3] showed that the problem can be solved in polynomial time.

Our results. In this paper we extend the results from [1, 3] as outlined in the previous two paragraphs to the SRI case. In particular, we consider the problem of finding a matching M with the minimum number of blocking pairs, given an SRI instance. Note that there is no stipulation on the size of M here: in view of previous hardness results [1, 3, 8], we already know that the problem is NP-hard and difficult to approximate if M is required to be of maximum cardinality. Rather, our assumption is that the stability of the matching is the overriding priority, and in cases where stability cannot be achieved, we wish to minimise the amount of "instability". Moreover we focus on the case that the length of the preference lists in a given SRI instance are bounded by some integer d. This reflects the fact that preference lists are often short in practical applications: for example a given kidney patient is likely to be compatible with only a relatively small subset of the available donors.

Let MIN BP d-SRI denote the problem of finding a matching with the minimum number of blocking pairs, given an SRI instance where all preference lists are of length at most d ($d \geq 1$). The main results in this paper are as follows:

1. for $d = 3$, MIN BP d-SRI is NP-hard not approximable within c, for some $c > 1$ unless P=NP;

2. for $d = 2$, MIN BP d-SRI is solvable in $O(m)$ time;

3. for $d \geq 3$, there is a straightforward $(2d-2)$-approximation algorithm for MIN BP d-SRI;

4. for $d \geq 3$, MIN BP d-SRI is approximable within $2d-3$. This performance guarantee improves to $2d-4$ if the instance admits no *elitist odd party*.

With respect to Result 4, an *elitist odd party*, which will be defined formally in Section 5, is a set of agents $\{a_0, a_1, \ldots, a_{k-1}\}$, for some odd $k \geq 3$, such that, for each i ($0 \leq i \leq k-1$), a_{i+1} and a_{i-1} are the first and second entries on a_i's preference list respectively, where addition and subtraction are taken modulo k. The definition of an elitist odd party is quite tightly constrained, and thus we would expect the majority of SRI instances not to admit such a structure. In such cases, the improved $2d-4$ performance guarantee prevails.

In the case that $d = 3$, our upper bound for the approximability of MIN BP d-SRI is 2 for instances with no elitist odd party, which increases to 3 if an elitist odd party does exist. Either case represents a substantial improvement over the performance guarantee of 4 as given by the straightforward $(2d-2)$-approximation algorithm. Another strength of our approach is that the approximation algorithm is valid for all $d \geq 3$, albeit with a performance guarantee that increases with d. On the other hand our lower bound for the hardness of approximating MIN BP d-SRI is quite close to 1 (~ 1.000984), suggesting that future work should address closing the gap between the lower and upper bounds.

Structure of the paper. The remainder of this paper is organised as follows. In Section 2, we define some important notation and terminology that will be used in the remainder of the paper. Then in Section 3, we give the lower bound for the approximability of MIN BP d-SRI, which holds even if $d = 3$. The simple linear-time algorithm for MIN BP 2-SRI is given in Section 4. For general $d \geq 3$, in Section 5 we give the approximation algorithm with performance guarantee $2d-3$, which improves to $2d-4$ in the absence of an elitist odd party. Finally, Section 6 contains some concluding remarks.

2 Preliminaries

We begin with a definition of the Stable Roommates problem with Incomplete lists (SRI). An instance I of SRI consists of an undirected graph $G = (A, E)$ where $A = \{a_1, \ldots, a_n\}$ and $m = |E|$. We assume that G contains no isolated vertices. The vertices of G are sometimes referred to as the *agents* of I. The vertices adjacent to a given agent a_i are the *acceptable agents* for a_i, and if $\{a_i, a_j\} \in E$, we say that a_i and a_j find each other *acceptable*. The input of I also contains a *preference list* associated with each agent $a_i \in A$, which is a total ordering of the vertices adjacent to a_i. We say that a_i *prefers* an agent a_j to another agent a_k if a_j precedes a_k in a_i's preference list. For a matching $M \subseteq E$ of I, if $\{a_i\ a_j\} \in M$ then $M(a_i)$ denotes a_j. A *blocking pair* is an edge $\{a_i, a_j\} \in E \backslash M$ such that (i) either a_i is unmatched in M, or a_i is matched in M and prefers a_j to $M(a_i)$, and (ii) either a_j is unmatched in M, or a_j is matched in M and prefers a_i to $M(a_j)$. Let $bp(M)$ denote the set of blocking pairs with respect to M in I. A matching M is *stable* if $bp(M) = \emptyset$. We also denote by $bp(I)$ the minimum value of $|bp(M')|$, taken over all matchings M' in I.

The classical Stable Marriage problem with Incomplete lists (SMI) [6, 7] is the special case of SRI in which the underlying graph G is bipartite. Moreover, in the special case of SRI that n is even and $m = n(n-1)/2$ (i.e., each agent finds all other agents acceptable), we obtain the Stable Roommates problem (SR).

3 Inapproximability for $d = 3$

In this section we show that MIN BP 3-SRI is NP-hard and not approximable within some $c > 1$ unless P=NP. To prove this, we give a reduction from a restricted version of SAT. Given a Boolean formula B in CNF and a truth assignment f, let $t(f)$ denote the number of clauses of B satisfied simultaneously by f, and let $t(B)$ denote the maximum value of $t(f)$, taken over all truth assignments f of B. Let MAX (2,2)-E3-SAT [2] denote the problem of finding, given a Boolean formula B in CNF in which each clause contains exactly 3 literals and each variable occurs exactly twice as an unnegated literal in B and exactly twice as a negated literal in B, a truth assignment f such that $t(f) = t(B)$.

Theorem 1. *Given any δ $(0 < \delta \leq \frac{1}{1016})$, MIN BP 3-SRI is NP-hard and not approximable within $\frac{1017}{1016} - \delta$ unless P=NP.*

Proof. Let B be an instance of MAX (2,2)-E3-SAT. Let $V = \{v_1, v_2 \ldots, v_n\}$ and $C = \{c_1, c_2, \ldots, c_m\}$ be the set of variables and clauses in B respectively. Then for each $v_i \in V$, each of literals v_i and $\bar{v}_i$ appears exactly twice in B. Also $|c_j| = 3$ for each $c_j \in C$. We form an instance I of MIN BP 3-SRI as follows.

Let $A_j = \{a_j^s : 1 \leq s \leq 3\}$, $B_j = \{b_j^s : 1 \leq s \leq 3\}$, $P_j = \{p_j^s : 1 \leq s \leq 3\}$ and $Q_j = \{q_j^s : 1 \leq s \leq 3\}$ define new sets of agents. For each clause $c_j \in C$, we create a gadget C_j in I containing the 20 agents in $A_j \cup B_j \cup P_j \cup Q_j \cup \{x_j^s, y_j^s : 1 \leq s \leq 4\}$. Also for each variable $v_i \in V$, we create a gadget V_i in I containing 4 agents $\{v_i^r : 1 \leq r \leq 4\}$. We refer to $A_j \cup B_j \cup P_j \cup Q_j$ as the set of *proper agents* in C_j (inducing the *proper part* of C_j), and the remaining agents in C_j are called *additional agents* (inducing the *additional part* of C_j). Finally, for each i $(1 \leq i \leq n)$, let $T_i = \{\{v_i^1, v_i^2\}, \{v_i^3, v_i^4\}\}$ and $F_i = \{\{v_i^1, v_i^4\}, \{v_i^2, v_i^3\}\}$.

The preference lists of these agents are shown in Figure 1, and this part of the instance is also illustrated in Figure 2.

In the list of each $a_j^s \in A_j$, if literal v_i appears at position s of clause $c_j \in C$, the symbol $v(a_j^s)$ denotes agent $v_i^{2(r-1)+1}$ where $r = 1, 2$ depending on whether this is the

$a_j^1:\ b_j^1\ \ v(a_j^1)\ \ q_j^1$
$a_j^2:\ b_j^2\ \ v(a_j^2)\ \ q_j^1$
$a_j^3:\ b_j^3\ \ v(a_j^3)\ \ q_j^3$
$p_j^1:\ b_j^1\ \ b_j^2\ \ p_j^2$
$p_j^2:\ p_j^1\ \ p_j^3$
$p_j^3:\ p_j^2\ \ b_j^3\ \ y_j^1$

$b_j^1:\ a_j^1\ \ p_j^1$
$b_j^2:\ a_j^2\ \ p_j^1$
$b_j^3:\ a_j^3\ \ p_j^3$
$q_j^1:\ a_j^1\ \ a_j^2\ \ q_j^2$
$q_j^2:\ q_j^1\ \ q_j^3$
$q_j^3:\ q_j^2\ \ a_j^3\ \ x_j^1$

$x_j^1:\ q_j^3\ \ x_j^2$
$x_j^2:\ x_j^1\ \ x_j^3\ \ x_j^4$
$x_j^3:\ x_j^4\ \ x_j^2$
$x_j^4:\ x_j^2\ \ x_j^3$
$y_j^1:\ p_j^3\ \ y_j^2$
$y_j^2:\ y_j^1\ \ y_j^3\ \ y_j^4$
$y_j^3:\ y_j^4\ \ y_j^2$
$y_j^4:\ y_j^2\ \ y_j^3$

$v_i^1:\ v_i^2\ \ a(v_i^1)\ \ v_i^4$
$v_i^3:\ v_i^4\ \ a(v_i^3)\ \ v_i^2$

$v_i^2:\ v_i^3\ \ a(v_i^2)\ \ v_i^1$
$v_i^4:\ v_i^1\ \ a(v_i^4)\ \ v_i^3$

Figure 1: Preference lists in the constructed instance of MIN BP 3-SRI.

first or second occurrence of literal v_i in B, otherwise if literal $\bar{v_i}$ appears at position s of clause $c_j \in C$, $v(a_j^s)$ denotes agent v_i^{2r} where $r = 1, 2$ depending on whether this is the first or second occurrence of literal $\bar{v_i}$ in B. Similarly, in the preference list of agent v_i^r for $r \in \{1, 3\}$, $a(v_i^r)$ denotes the agent a_j^s such that the $(\frac{r+1}{2})$th occurrence of v_i appears at position s of c_j. Finally, in the preference list of agent v_i^r for $r \in \{2, 4\}$, $a(v_i^r)$ denotes the agent a_j^s such that the $(\frac{r}{2})$th occurrence of $\bar{v_i}$ appears at position s of c_j. Note that $v(a_j^s) = v_i^r$ if and only if $a(v_i^r) = a_j^s$.

We claim that $t(B) + bp(I) = 2m$. To show that $bp(I) \leq 2m - t(B)$, suppose that we are given a truth assignment f with $t(f) = t(B)$; we create a matching M in I such that $|bp(M)| = 2m - t(f)$. For each variable $v_i \in V$, if v_i is true under f, add the pairs in T_i to M, otherwise add the pairs in F_i to M.

Now let $c_j \in C$. If c_j contains a literal that is true under f, let $s \in \{1, 2, 3\}$ denote the position of c_j in which this literal occurs, otherwise set $s = 1$. Add the pairs (a_j^t, b_j^t) $(1 \leq t \neq s \leq 3)$ to M, and match the rest of the proper part of C_j in the only way such that no proper agent is unmatched in M, namely, for

- $s = 1$, add $M_j^1 = \{\{a_j^1, q_j^1\}, \{b_j^1, p_j^1\}, \{q_j^2, q_j^3\}, \{p_j^2, p_j^3\}\}$ to M;
- $s = 2$, add $M_j^2 = \{\{a_j^2, q_j^1\}, \{b_j^2, p_j^1\}, \{q_j^2, q_j^3\}, \{p_j^2, p_j^3\}\}$ to M;
- $s = 3$, add $M_j^3 = \{\{a_j^3, q_j^3\}, \{b_j^3, p_j^3\}, \{p_j^1, p_j^2\}, \{q_j^1, q_j^2\}\}$ to M.

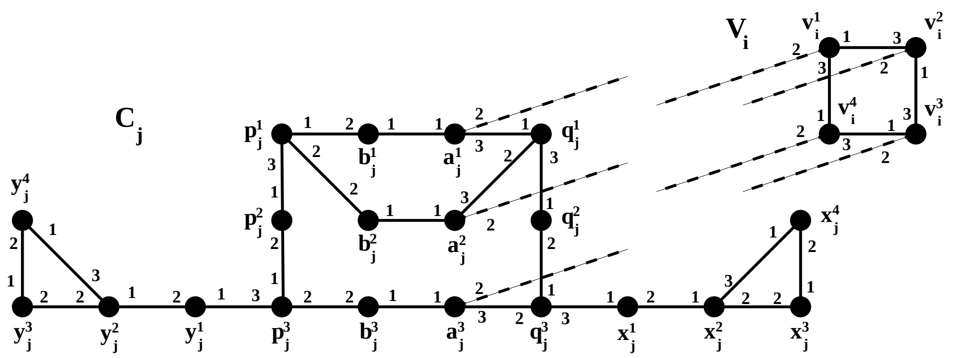

Figure 2: Gadgets C_j and V_i, where the dashed lines represent the interconnecting edges. The preferences of a given agent a are shown by annotating the edges incident to a.

Finally, add $\{\{x_j^1, x_j^2\}, \{x_j^3, x_j^4\}, \{y_j^1, y_j^2\}, \{y_j^3, y_j^4\}\}$ to M for each j ($1 \leq j \leq m$). Observe that $bp(M) \cap (A_j \times B_j) = \{a_j^s, b_j^s\}$. Now if c_j is not satisfied under f then agent $v(a_j^1)$ has her last-choice partner, by construction of M. Hence $\{v(a_j^1), a_j^1\} \in bp(M)$. Moreover these, together with the m blocking pairs identified already, are all the blocking pairs of M in I. Hence $|bp(M)| = m + (m - t(f))$, as required. (Note that M is a perfect matching in I.)

To show that $bp(I) \geq 2m - t(B)$, suppose for a contradiction that there is a matching M with $|bp(M)| = bp(I) < 2m - t(B)$. The most important point of our argument is that M can be chosen to be perfect, since we can prove that if M is not perfect then we can create a perfect matching M^* with $|bp(M^*)| \leq |bp(M)|$ as follows.

First we show that, given a matching M, we can create a matching M' with $|bp(M')| \leq |bp(M)|$ such that M' covers all the additional agents. Consider a set of linked additional agents, say $\{x_j^1, x_j^2, x_j^3, x_j^4\}$, and suppose that not all of them are covered in M. If $\{x_j^1, q_j^3\} \notin M$ then we can clearly match all of these additional agents without introducing any new blocking pairs. If $\{x_j^1, q_j^3\} \in M$ then add $\{\{x_j^1, x_j^2\}, \{x_j^3, x_j^4\}\}$ to M', by leaving q_j^3 unmatched in M', and let M' be the same as M for the rest of the instance. Here $\{q_j^3, x_j^1\} \in bp(M') \backslash bp(M)$, however, since one of the edges from $\{\{x_j^2, x_j^3\}, \{x_j^3, x_j^4\}, \{x_j^4, x_j^2\}\}$ is in $bp(M) \backslash bp(M')$, the number of blocking pairs remains the same. After making these changes for each component of additional agents, we obtain a matching M' that covers every additional agent and satisfies $|bp(M')| \leq |bp(M)|$.

In the second step, we remove all *interconnecting edges*, i.e. edges of the form $\{a_j^s, v_i^r\}$, from M' and we rearrange matching M' in every gadget that admits an agent who is covered by an interconnecting edge in M'. We call these gadgets *affected gadgets*. In each affected gadget C_j, let $\{\{a_j^s, b_j^s\} : 1 \leq s \leq 3\} \cup \{\{q_j^1, q_j^2\}, \{p_j^1, p_j^2\}\}$ belong to the new matching M'', leaving p_j^3 and q_j^3 unmatched. In each affected gadget V_i, if $\{v_i^1, v_i^4\}$ or $\{v_i^2, v_i^3\}$ is in M' then add $\{\{v_i^1, v_i^4\}, \{v_i^2, v_i^3\}\}$ to M'', otherwise add $\{\{v_i^1, v_i^2\}, \{v_i^3, v_i^4\}\}$ to M''.

To show that $|bp(M'')| \leq |bp(M')|$, first we observe that no interconnecting edge can be blocking for M'' if it was not blocking for M'. The interconnecting edges of the form $\{a_j^s, v_i^r\}$, where C_j is affected, cannot be blocking, since each a_j^s is matched to her best partner in M''. The interconnecting edges of the form $\{a_j^s, v_i^r\}$, where V_i is affected but C_j is not affected, cannot belong to $bp(M'') \backslash bp(M')$, since each v_i^r has either remained matched to the same partner in M'' or was unmatched in M' (and the partner of a_j^s has not changed). Now we prove that the number of blocking pairs has not increased within any of the affected gadgets. Clearly, an affected V_i does not admit any blocking pair for M''. An affected C_j admits exactly two blocking pairs for M'' (i.e., $\{x_j^1, q_j^3\}$ and $\{y_j^1, p_j^3\}$), so we shall prove that C_j admits at least two blocking pairs for M' too. If $\{a_j^s, v_i^r\} \in M'$ then $\{a_j^s, b_j^s\} \in bp(M')$. Furthermore, if $\{a_j^t, b_j^t\} \notin M'$ for some ($1 \leq t \neq s \leq 3$) then $\{a_j^t, b_j^t\} \in bp(M')$. On the other hand, if $\{a_j^t, b_j^t\} \in M'$ for each ($1 \leq t \neq s \leq 3$) then at least one proper agent from $\{q_j^1, q_j^3\}$ must be uncovered in M' which induces another blocking pair, since each of these agents is the first choice of somebody else. Therefore $|bp(M'')| \leq |bp(M')|$, as claimed.

In the third and final step, we modify M'' and obtain a perfect matching M^* such that $|bp(M^*)| \leq |bp(M'')|$. Note that for each gadget C_j ($1 \leq j \leq m$), there are three matchings that cover the proper part of C_j, namely $\bar{M}_j^s$ for ($1 \leq s \leq 3$), where $\bar{M}_j^s = M_j^s \cup \{\{a_j^t, b_j^t\} : 1 \leq t \neq s \leq 3\}$. For each $\bar{M}_j^s$ ($1 \leq s \leq 3$), there is exactly one blocking pair involving two agents of C_j (i.e., $\{a_j^s, b_j^s\}$), and at most one further blocking pair involving one agent from C_j (i.e., possibly $\{a_j^s, v(a_j^s)\}$). On the other hand, if M'' does not cover every proper agent in C_j, then at least two proper agents are uncovered, since M'' contains no interconnecting edges. Note that these agents cannot be adjacent to each other, since otherwise the number of blocking pairs could not be minimum in M'', obviously. Furthermore, every proper agent in C_j is somebody else's first choice, so M'' must admit at least two blocking pairs within C_j. Therefore, in the latter case, we can replace the restriction of M'' to C_j by any of $\{\bar{M}_j^s : 1 \leq s \leq 3\}$ without increasing the number of blocking pairs. Finally, if the restriction of M'' does not cover every agent in V_i then we can always extend it to obtain either T_i or F_i without creating any new blocking pair.

As a result, we obtain a perfect matching M^*, for which $|bp(M^*)| < 2m - t(B)$. We also know that the restriction of M^* to the proper part of C_j ($1 \leq j \leq m$) is $\bar{M}_j^s$ for some s ($1 \leq s \leq 3$) and the restriction of M^* to V_i ($1 \leq i \leq n$) is either T_i or F_i. Moreover for each j ($1 \leq j \leq m$), $\{a_j^s, b_j^s\} \in bp(M^*)$ for some s ($1 \leq s \leq 3$). Additionally, $bp(M^*)$ contains pairs of the form $\{a_j^s, v(a_j^s)\}$. Now let f be a truth assignment of B such that each variable v_i is true if and only if the restriction of M^* to V_i is T_i. Clearly, a pair of form $\{a_j^s, v(a_j^s)\} \in bp(M^*)$ if and only if the literal occurring at position s of c_j is false, therefore

$$t(f) \geq 2m - |bp(M^*)| > 2m - (2m - t(B)) = t(B),$$

a contradiction. Hence we proved that $t(B) + bp(I) = 2m$.

Now let $\varepsilon = \delta/3$. Berman et al. [2] show that it is NP-hard to distinguish between instances B of MAX (2,2)-E3-SAT for which (i) $t(B) \geq (1-\varepsilon)m$ and (ii) $t(B) \leq \left(\frac{1015}{1016} + \varepsilon\right) m$. By our construction, it follows that in case (i), $bp(I) \leq (1+\varepsilon)\, m$, whilst in case (ii), $bp(I) \geq \left(\frac{1017}{1016} - \varepsilon\right) m$. Hence an approximation algorithm for MIN BP 3-SRI with performance guarantee $\frac{1017}{1016} - \delta < \frac{\frac{1017}{1016}-\varepsilon}{1+\varepsilon}$ could be used to decide between cases (i) and (ii) for MAX (2,2)-E3-SAT in polynomial time, which is a contradiction unless P=NP. □

4 Algorithm for $d \leq 2$

Tan [15] defined a *stable partition* in a given instance I of SRI, which is a generalization of the concept of a stable matching. We will utilise this concept in this section and subsequently. Pittel and Irving [13] gave a concise definition of a stable partition, however their definition requires that each agent a_i ranks himself last on his preference list, after all of the other agents on his list (this constitutes a self-loop in the underlying graph G). We assume that this is the case in presenting the following definition.

Definition 2 ([13]). *Let I be an* SRI *instance where A is the set of agents. A* stable partition *is a permutation Π of A satisfying the following two properties:*

1. *for each $a_i \in A$, either $\Pi(a_i) = \Pi^{-1}(a_i)$ or a_i prefers $\Pi(a_i)$ to $\Pi^{-1}(a_i)$;*
2. *if a_i prefers a_j to $\Pi^{-1}(a_i)$, then a_j prefers $\Pi^{-1}(a_j)$ to a_i.*

We refer to a cycle in Π with odd (respectively, even) length as an odd *(respectively,* even*)* party *of Π.*

Note that, possibly a_i is a fixed point of Π, so that $\Pi(a_i) = \Pi^{-1}(a_i) = a_i$. We consider this to be an odd party of size one. The following theorem regarding stable partitions is due to Tan [15, 16].

Theorem 3 ([15, 16]). *Let I be an instance of* SRI. *Then I admits at least one stable partition, which can be found in $O(m)$ time, where m is the number of acceptable pairs in I. Furthermore, any two stable partitions in I have exactly the same set of odd parties. Finally, I admits a stable matching if and only if Π has no odd party of size ≥ 3.*

Suppose we are given an instance I of MIN BP 2-SRI. Clearly the connected components of the underlying graph G are paths and cycles. Construct a stable partition Π in I. Paths and even-length cycles in G are bipartite, and hence Π gives rise to a stable matching within each such component. Now consider each odd-length cycle C in G. If Π induces an odd party P of size ≥ 3 in C, then by deleting an agent from P and forming a perfect matching among the edges that remain, we obtain a matching in C with one blocking pair. Otherwise Π induces a stable matching in C. After considering each component in G in turn, we thus arrive at a matching with the minimum number of blocking pairs. We summarise this discussion with the following theorem.

Theorem 4. MIN BP *2*-SRI *is solvable in $O(m)$ time, where m is the number of acceptable pairs in a given instance I. Moreover $bp(I)$ is equal to the number of odd parties of size ≥ 3 in a stable partition in I.*

5 Approximation algorithm for $d \geq 3$

5.1 Preliminary results

We mentioned in the introduction that Tan [16] gave an $O(m)$ algorithm for finding the smallest set of agents that need to be deleted from a given SRI instance I in order to leave a stable matching. Tan's algorithm is based on finding a stable partition Π in I, and attempting to match as many agents as possible within their own party in Π in the following way. For each even party $P = (a_0, a_1, \ldots, a_{2k-1})$ $(k \geq 1)$, we match $\{a_{2i}, a_{2i+1}\}$ for all i $(0 \leq i \leq k-1)$. For odd parties P, we select an arbitrary agent $a_i \in P$ to delete from the instance, decompose $P - \{a_i\}$ into a maximum set of pairs, and add these pairs to the matching. When P is an even party, or P is an odd party with an agent $a_i \in P$, let us refer to this process of decomposing P (respectively, $P \backslash \{a_i\}$) into a maximum set of matched pairs as *decomposing and matching* P (respectively, $P \backslash \{a_i\}$). The following lemma, and its immediate corollary illuminates why this approach may also be a good approach for MIN BP d-SRI.

Lemma 5. *Let M' be any matching for I. Then, an agent a_i matched to either $\Pi(a_i)$ or $\Pi^{-1}(a_i)$ does not block with an agent a_j matched to either $\Pi(a_j)$ or $\Pi^{-1}(a_j)$. Consequently, an agent a_i in a party of size one does not block with an agent a_j matched to either $\Pi(a_j)$ or $\Pi^{-1}(a_j)$, regardless of to whom a_i is matched.*

Proof. Recall property (1) of the stable partition, which states that either an agent a_i prefers $\Pi(a_i)$ to $\Pi^{-1}(a_i)$ or $\Pi(a_i) = \Pi^{-1}(a_i)$. By property (2) of the stable partition, if an agent a_i prefers another agent a_j to $\Pi^{-1}(a_i)$, then a_j does not prefer a_i to $\Pi^{-1}(a_j)$. Hence, if a_i is matched to either $\Pi^{-1}(a_i)$ or $\Pi(a_i)$, he cannot be in a blocking pair with a_j if a_j is matched to $\Pi^{-1}(a_j)$ or $\Pi(a_j)$. Now, translate this into the context of a_i being in an odd party of size one: if a_i is unmatched (i.e., he is matched to $\Pi(a_i) = \Pi^{-1}(a_i) = a_i$), then he does not block with any agent a_j matched to $\Pi(a_j)$ or $\Pi^{-1}(a_j)$. Hence, if a_i is matched to anyone, he still cannot block with a_j. $\square$

Corollary 6. *Let M' be any matching for I. Then, for every blocking pair $\{a_i, a_j\}$ relative to M', at least one of $\{a_i, a_j\}$, say, a_i, is not matched to $\Pi(a_i)$ or $\Pi^{-1}(a_i)$.*

This corollary leads one to believe that a reasonable attempt at minimizing blocking pairs is indeed to simply decompose and match each party, excluding an arbitrary agent (or, even better, one of minimum degree) from each odd party. This, in fact, does give a constant performance guarantee as follows. Let $\mathcal{P}$ denote the set of all parties in Π, and $\mathcal{P}_O$ the set of odd parties of size ≥ 3. Given $P_i \in \mathcal{P}_O$, let $d_i = \min_{a_j \in P_i} d_G(a_j)$, where

$d_G(a_j)$ denotes the degree of vertex a_j in the underlying graph G. Abraham et al. [1] showed that the following upper and lower bounds hold for $bp(I)$.

Proposition 7 ([1]). $\left\lceil \frac{|\mathcal{P}_O|}{2} \right\rceil \quad \leq \quad bp(I) \quad \leq \quad \sum_{P_i \in \mathcal{P}_O} (d_i - 1).$

The latter upper bound is achieved by using Tan's algorithm, where, for each odd party P_i of size ≥ 3, we choose an agent $a_k \in P_i$ having minimum degree in G amongst all agents from P to be unmatched. By Lemma 5, a_k cannot block with $\Pi(a_k)$, and therefore can be in at most $d_i - 1$ blocking pairs. When the preference lists have length at most d, for some $d \geq 3$, the upper bound can be set to be $(d-1)|\mathcal{P}_O|$. Thus, this use of Tan's algorithm leads us to a straightforward $(2d-2)$-approximation of MIN BP d-SRI.

We now show how to improve on this performance guarantee for MIN BP d-SRI. Our improved algorithm achieves a superior performance guarantee by very selectively deciding which agents from each odd party will be excluded from being matched within their party, and either match them with another excluded agent, or decide that they will be unmatched. We crucially rely on the properties of a particular type of odd party relative to Π, which we call an *elitist odd party*, defined as follows.

Definition 8. *An* elitist odd party *is an odd party $P = (a_0, a_1, \ldots, a_k)$ in Π with $k \geq 2$ such that $\Pi(a_i)$ and $\Pi^{-1}(a_i)$ are the first and second entries, respectively, of a_i's preference list for $0 \leq i \leq k$.*

Note that by Theorem 3, the definition of an elitist odd party is independent of the particular stable partition in I chosen. Recall that $\mathcal{P}$ and $\mathcal{P}_O$ are the set of all parties and odd parties of size ≥ 3, respectively, in Π. We further denote the set of even parties, odd parties of size one, and elitist odd parties in $\mathcal{P}$ by $\mathcal{P}_E$, $\mathcal{P}_O^1$, and $\mathcal{P}_O^e$, respectively. All other odd parties not in $\mathcal{P}_E \cup \mathcal{P}_O^1 \cup \mathcal{P}_O^e$, which must therefore be non-elitist odd parties of size ≥ 3, are denoted by $\mathcal{P}_O^{ne}$. For a given agent a_i, we let $P(a_i)$ denote the party of a_i. Also, for any party P in Π, $A(P)$ denotes the set of agents in P. It is important to note the following remark, which follows immediately from property 2 of the definition of a stable partition.

Remark. The set of vertices in the set $\mathcal{P}_O^1$ of odd parties of size one in $\mathcal{P}$ constitutes an independent set in G.

5.2 Approximation algorithm

The approximation algorithm, which is given in Figure 3, takes a four-phase approach to compute a matching M for I.

Before the first phase, we set the stage by computing an arbitrary stable partition Π, and by setting the matching M to be returned by the algorithm to be the empty set. Each odd party is defined to be *undestroyed* – loosely speaking, this terminology means that we have not yet decided how these agents will be matched in M.

Phase one: even parties are easy.
In phase one, all even parties are decomposed and matched as described immediately prior to Lemma 5. These pairs are added to the matching M.

Phase two: pair together as many odd parties as possible – with a twist.
How shall we decide what agent to exclude from each odd party? Consider the following simple observation. If $P_i = (a_0, a_1, \ldots, a_k)$ and $P_j = (b_0, b_1, \ldots, b_l)$ are odd parties such that $\{a_r, b_s\}$ is an edge of G, with $a_r \in P_i$ and $b_s \in P_j$, then, we could match $\{a_r, b_s\}$ and decompose and match $P_i \backslash \{a_r\}$ and $P_j \backslash \{b_s\}$. (Note that, if one of these two odd parties,

Procedure MIN BP d-SRI**-approx**:
compute Π
$M \leftarrow \emptyset$
set every odd party in $\mathcal{P}_O^1 \cup \mathcal{P}_O^e \cup \mathcal{P}_O^{ne}$ to be *undestroyed*

phase one:
for each even party P in Π:
    decompose and match P, and add these pairs to M

phase two:
construct the auxiliary graph H
$M_H \leftarrow$ maximum matching in H
$M'_H \leftarrow \emptyset$
for each pair $\{P_i, P_j\}$ in M_H:
    add one acceptable pair $\{a_r, a_s\}$ with $a_r \in P_i$ and $a_s \in P_j$ to M'_H
    decompose and match $P_i \backslash \{a_r\}$ and $P_j \backslash \{a_s\}$, and add these pairs to M
    set P_i and P_j to be destroyed

$U_{\mathcal{P}} \leftarrow$ set of unmatched parties relative to M_H
$U_A \leftarrow$ set of agents in the parties in $U_{\mathcal{P}}$
while $\exists \{a_i, a_j\} \in M'_H$ and a set $B = \{b_1, \dots, b_t\} \subseteq U_A$ such that, for each $b_l \in B$, $\{a_i, b_l\}$ forms a blocking pair relative to M'_H:
    $b_k \leftarrow a_i$'s most preferred agent from $\{b_1, \dots, b_t\}$
    $M_H \leftarrow (M_H \backslash \{\{P(a_i), P(a_j)\}\}) \cup \{\{P(a_i), P(b_k)\}\}$
    $M'_H \leftarrow (M'_H \backslash \{\{a_i, a_j\}\}) \cup \{\{a_i, b_k\}\}$
    decompose and match $P(b_k) \backslash \{b_k\}$, and add these pairs to M
    set $P(b_k)$ to be destroyed
    $U_{\mathcal{P}} \leftarrow U_{\mathcal{P}} \backslash \{P(b_k)\}$
    remove the agents of $P(b_k)$ from U_A // no agents are added to U_A
add all pairs in M'_H to M

phase three:
while $\exists$ a_i in an undestroyed non-elitist odd party P with $|P| \geq 3$:
    $a_i \leftarrow$ arbitrary agent in P that prefers some $a_j \neq \Pi(a_i)$ to $\Pi^{-1}(a_i)$
    decompose and match $P \backslash \{a_i\}$, and add these pairs to M
    set P to be destroyed

phase four:
while $\exists$ an undestroyed elitist odd party P:
    $a_i \leftarrow$ arbitrary agent in P
    decompose and match $P \backslash \{a_i\}$, and add these pairs to M
    set P to be destroyed

Figure 3: A pseudocode description of the approximation algorithm.

P_i say, is an odd party of size one, then nothing happens when we decompose and match $P_i \backslash \{a_r\}$.) Rather than myopically selecting two such odd parties, in phase two we will compute a maximum pairing of these odd parties, and then adjust this pairing for our purposes.

We construct an auxiliary graph H. The vertex set of H is the set of odd parties, i.e., $(\mathcal{P}_O^{ne} \cup \mathcal{P}_O^1 \cup \mathcal{P}_O^e)$. We sometimes refer to the vertices of this graph as parties. The edges of H are defined to be all pairs of odd parties $\{P_i, P_j\}$ such that there exists an acceptable pair $\{a_r, a_s\}$ with $a_r \in P_i$ and $a_s \in P_j$. We compute a maximum matching M_H for H, and construct an additional matching M'_H as follows. For each pair $\{P_i, P_j\}$ in M_H, arbitrarily choose exactly one acceptable pair $\{a_r, a_s\}$ to add to M'_H where $a_r \in P_i$ and $a_s \in P_j$ (such a pair exists, otherwise $\{P_i, P_j\}$ is not an edge of H). Next, decompose and match $P_i \backslash \{a_r\}$ and $P_j \backslash \{a_s\}$, and add these pairs to M – but note that $\{a_r, a_s\}$ is not added to M. These parties are now destroyed.

Next, we adjust the matchings M_H and M'_H. Let $U_{\mathcal{P}}$ denote the set of unmatched parties relative to M_H, and U_A the set of agents in $U_{\mathcal{P}}$. While there exists an edge $\{a_i, a_j\} \in M'_H$ such that there is a set of agents $\{b_1, \ldots, b_t\} \in U_A$ that form a blocking pair (relative to M'_H) with a_i, let b_k denote a_i's most preferred agent from $\{b_1, \ldots, b_t\}$. Remove the pair $\{P(a_i), P(a_j)\}$ from M_H, and replace it with $\{P(a_i), P(b_k)\}$. Correspondingly, remove the pair $\{a_i, a_j\}$ from M'_H, and replace it with $\{a_i, b_k\}$. Next, decompose and match $P(b_k)$, and add these pairs to M. This party is now destroyed. The set $U_{\mathcal{P}}$ now contains one fewer odd party, and the set U_A at least one fewer agent – hence the loop terminates. Crucially, the agent a_j is not added to U_A, nor is his party added to $U_{\mathcal{P}}$ (it has already been decomposed and matched). When the loop ends, all pairs in M'_H are added to M.

We remark on two important subtleties of the while loop of phase two. First, if a pair $\{a_i, a_j\}$ is replaced with the pair $\{a_i, b_k\}$, then this pair will remain in the matching M'_H until the end of the while loop. This is proven in Lemma 9, where we show that it turns out that no agent in the set U_A at any further iteration of the loop can block with a_i. It also not immediately obvious (but nevertheless true) that no agent in U_A at any further iteration of the loop finds a_j or b_k acceptable. This is also shown in Lemma 9.

Phase three: destroy the remaining non-elitist odd parties of size ≥ 3.
Phase three consists of a loop that continues while there exists a non-elitist undestroyed odd party P with $|P| \geq 3$. By definition, P contains an agent a_i with some $a_j \neq \Pi(a_i)$ on his list that he prefers to $\Pi^{-1}(a_i)$. We identify this a_i and decompose and match $P \backslash \{a_i\}$, and add these pairs to M (hence, a_i is unmatched in M). This party is now considered destroyed. It turns out (proven in Lemma 10) that a_i can never block with a_j.

Phase four: destroy the remaining elitist odd parties.
Finally, phase four iteratively considers each remaining undestroyed elitist odd party P. An arbitrary agent $a_i \in P$ is selected, and $P \backslash \{a_i\}$ is decomposed and matched. This party is now considered destroyed. The algorithm then returns the matching M.

We remark that, in general, some elitist odd parties may be destroyed prior to phase four. Also, some odd parties of size 1 may end up being undestroyed. For our purposes, it is irrelevant as to whether either of these occurs.

Next, we illustrate the execution of the approximation algorithm on an instance I of SRI with $d = 3$.

Example

Let the preference lists of the agents be as shown in Figure 4.

Here, the unique stable partition is $\Pi = (a_1, a_2, a_3)(a_4, a_5)(a_6, a_7, a_8)(a_9)(a_{10}, a_{11}, a_{12})$, where $P_1 = (a_1, a_2, a_3)$ is an elitist odd party, $P_2 = (a_4, a_5)$ is an even party, $P_3 = (a_6, a_7, a_8)$ and $P_5 = (a_{10}, a_{11}, a_{12})$ are non-elitist odd parties and $P_4 = (a_9)$ is an odd party of size one.

In the first phase of the algorithm we match a_4 with a_5. In the second phase we construct the auxiliary graph H that is a triangle consisting of P_1, P_3 and P_4 as vertices and

$a_1:$	a_2	a_3	a_8	$a_5:$	a_4	a_6	a_{11}	$a_9:$	a_7	a_2	
$a_2:$	a_3	a_1	a_9	$a_6:$	a_5	a_7	a_8	$a_{10}:$	a_{11}	a_{12}	a_4
$a_3:$	a_1	a_2	a_4	$a_7:$	a_8	a_6	a_9	$a_{11}:$	a_{12}	a_5	a_{10}
$a_4:$	a_3	a_{10}	a_5	$a_8:$	a_6	a_7	a_1	$a_{12}:$	a_{10}	a_{11}	

Figure 4: Preference lists in the instance of SRI with twelve agents.

an isolated vertex corresponding to P_5. Now the algorithm randomly chooses a maximum matching H, that is one edge from the three. We describe all the three possible cases.

Suppose first that pair $\{P_1, P_3\}$ is selected, so we match agents a_1 and a_8 in M'_H and we decompose and match $P_1 \backslash \{a_1\}$ and $P_2 \backslash \{a_8\}$ by adding $\{a_2, a_3\}$ and $\{a_6, a_7\}$ to the final matching M. The while loop in the second phase does not make any change in M'_H so we add $\{a_1, a_8\}$ to M. In phase three we take P_5 and we select a_{11} to be unmatched (as a_{11} prefers a_5 to $a_{10} = \Pi^{-1}(a_{11})$), and we decompose and match $P_5 \backslash \{a_{11}\}$ by adding $\{a_{10}, a_{12}\}$ to M. The matching M remains the same in the last phase, so the resulting matching is $M_1 = \{\{a_1, a_8\}, \{a_2, a_3\}, \{a_4, a_5\}, \{a_6, a_7\}, \{a_{10}, a_{12}\}\}$ with three blocking pairs, i.e., $bp(M_1) = \{\{a_1, a_3\}, \{a_7, a_8\}, \{a_{10}, a_{11}\}\}$.

Suppose now that the algorithm selects $\{P_1, P_4\}$ instead. We add $\{a_2, a_9\}$ to M'_H and we decompose and match $P_1 \backslash \{a_2\}$ by adding $\{a_1, a_3\}$ to the final matching M. In the while loop of the second phase, $\{a_7, a_9\}$ is a blocking pair relative to M'_H, such that a_7 is the most preferred agent for a_9 satisfying the requirements, so we adjust M_H by removing $\{P_1, P_4\}$ and adding $\{P_3, P_4\}$. We also remove $\{a_2, a_9\}$ from M'_H and we add $\{a_7, a_9\}$ instead, furthermore we decompose and match $P_3 \backslash \{a_7\}$ by adding $\{a_6, a_8\}$ to the final matching M. The while loop stops without any more changes so we add $\{a_7, a_9\}$ to M as well. In phase three we select a_{11} to be unmatched, and we decompose and match $P_5 \backslash \{a_{11}\}$ by adding $\{a_{10}, a_{12}\}$ to M. All the odd elitist parties are destroyed so phase four is skipped. The final matching is $M_2 = \{\{a_1, a_3\}, \{a_4, a_5\}, \{a_6, a_8\}, \{a_7, a_9\}, \{a_{10}, a_{12}\}\}$ with three blocking pairs, i.e., $bp(M_2) = \{\{a_1, a_2\}, \{a_6, a_7\}, \{a_{10}, a_{11}\}\}$.

Finally, in the third case $\{P_3, P_4\}$ is chosen. We add $\{a_7, a_9\}$ to M'_H and we decompose and match $P_3 \backslash \{a_7\}$ by adding $\{a_6, a_8\}$ to the final matching M. The while loop of the second phase runs without making any change to M'_H, so we add $\{a_7, a_9\}$ to M. In phase three again we select a_{11} to be unmatched, and we decompose and match $P_5 \backslash \{a_{11}\}$ by adding $\{a_{10}, a_{12}\}$ to M. In the fourth phase the remaining elitist cycle (a_1, a_2, a_3) is decomposed and matched in an arbitrary way, say by choosing a_3 to remain unmatched and by adding $\{a_1, a_2\}$ to M. The final matching in this case is $M_3 = \{\{a_1, a_2\}, \{a_4, a_5\}, \{a_6, a_8\}, \{a_7, a_9\}, \{a_{10}, a_{12}\}\}$ which admits four blocking pairs, i.e., $bp(M_3) = \{\{a_2, a_3\}, \{a_3, a_4\}, \{a_6, a_7\}, \{a_{10}, a_{11}\}\}$.

Note that there is a matching $M^* = \{\{a_1, a_2\}, \{a_3, a_4\}, \{a_5, a_6\}, \{a_7, a_8\}, \{a_{10}, a_{12}\}\}$ that admits two blocking pairs (i.e., $bp(M^*) = \{\{a_2, a_3\}, \{a_{10}, a_{11}\}\}$), which is optimal for this instance since every stable partition admits three odd parties of size at least three (recall from Proposition 7 that $bp(I) \geq \lceil |\mathcal{P}_O|/2 \rceil$, and $|\mathcal{P}_O| = 3$ in I).

Finally we also note that a modified example may be constructed to show that the upper bound $2d-3$ for the performance ratio of our approximation algorithm can be achieved in the case where the instance admits an elitist odd party. By removing agents a_{10}, a_{11} and a_{12} from the instance the new optimal matching $M^* = \{\{a_1, a_2\}, \{a_3, a_4\}, \{a_5, a_6\}, \{a_7, a_8\}\}$ would admit only one blocking pair (i.e., $bp(M^*) = \{\{a_2, a_3\}\}$), whilst the matching obtained in the third case according to the above argument adjusted to the reduced instance, $M_3 = \{\{a_1, a_2\}, \{a_4, a_5\}, \{a_6, a_8\}, \{a_7, a_9\}\}$ would admit three blocking pairs, i.e.,

$bp(M_3) = \{\{a_2, a_3\}, \{a_3, a_4\}, \{a_6, a_7\}\}$.

5.3 An upper bound on the blocking pairs

We require two lemmas, one regarding phase two, and the other regarding phase three, before presenting the main theorem of this section, which bounds the number of blocking pairs relative to the matching M returned by the algorithm.

Lemma 9. *Suppose that there is a pair $\{a_i, a_j\} \in M'_H$ and a set $B = \{b_1, \ldots, b_t\} \subseteq U_A$ such that, for each $b_l \in B$, $\{a_i, b_l\}$ forms a blocking pair relative to M'_H with a_i at a particular point in the execution of phase two.*

1. *If any agent a_k in U_A finds a_j acceptable, then a_k and all agents in B are in the same odd party.*
2. *If phase two of the algorithm replaces the pair $\{a_i, a_j\}$ with $\{a_i, b_k\}$ at this time, then, at any subsequent step in the execution of phase two, no agent in $U_A \backslash A(P(b_k))$ forms a blocking pair with a_i, relative to M'_H, and no agent in $U_A \backslash A(P(b_k))$ finds b_k or a_j acceptable.*
3. *The pair $\{a_i, b_k\}$ described in (2) is never removed from M'_H, therefore it is added to M at the end of phase two.*

Proof. (1). If an agent a_k in U_A finds a_j acceptable, and is not in the same odd party as some agent $b_l \in B$, then $P(a_k)$ - $P(a_j)$ - $P(a_i)$ - $P(b_l)$ is an augmenting path for M_H, contradicting that M_H is a maximum matching for H.

(2). Let $a_l \in U_A \backslash P(b_k)$. If, after replacing the pair $\{a_i, a_j\}$ with $\{a_i, b_k\}$, a_l blocks with a_i, then the choice of b_k was not valid, as b_k was chosen to be a_i's most preferred blocking agent. If a_l finds b_k acceptable, then there is an edge $\{P(a_l), P(b_k)\}$ in H. Since $P(a_l)$ and $P(b_k)$ were both in $U_{\mathcal{P}}$ prior to the pair $\{P(a_i), P(a_j)\}$ being removed from M_H, M_H cannot be a maximum matching. Lastly, it follows from (1) that a_l cannot find a_j acceptable, as he is in a different odd party than b_k, who was in B. As the set U_A only decreases with each iteration of the while loop, a_l still cannot block with a_i, and he clearly cannot change his preference list in order to find b_k or a_j acceptable.

(3). This follows immediately from the proof of 2. For the pair $\{a_i, b_k\}$ to be removed from M'_H, some agent a_l in the set U_A at a later iteration of the while loop has to form a blocking pair with either a_i or b_k, a contradiction. □

The next lemma identifies the facts we need regarding phase three of the algorithm.

Lemma 10. *Let P be a non-elitist odd party of size ≥ 3 that is undestroyed at the end of phase two.*

1. *There exists an agent $a_i \in P$ who prefers some $a_j \neq \Pi(a_i)$ to $\Pi^{-1}(a_i)$.*
2. *P is in the set $U_{\mathcal{P}}$ at the start and end of phase two.*
3. *If a_j is not in P, then a_j is matched in M, and does not block with a_i (relative to M).*

Proof. (1). This follows immediately from the definition of being a non-elitist odd party.

(2). Since P is undestroyed, it was never matched in M_H, and therefore it was in $U_{\mathcal{P}}$ at the end of phase two. Since the contents of U_P at the end of phase 2 form a proper subset of its contents at the start of phase 2, P was in $U_{\mathcal{P}}$ at the start of phase two as well.

(3). Suppose a_j is not in P, and that $\{a_i, a_j\}$ form a blocking pair relative to M at the moment the party P is selected by phase three. By property (2) of the stable partition, we know that since a_i prefers a_j to $\Pi^{-1}(a_i)$, a_j prefers $\Pi^{-1}(a_j)$ to a_i. Hence if a_j is matched to $\Pi^{-1}(a_i)$ or $\Pi^{-1}(a_j)$, he does not block with a_i. So, suppose that a_j is in a different odd party $P(a_j)$ (he cannot be in an even one, nor by assumption can he be in $P(a_i)$), and is not matched to $\Pi^{-1}(a_i)$ or $\Pi(a_i)$. Since M_H is maximum and (by part (2) of this lemma) $P(a_i)$ was never matched in M_H, $P(a_j)$ must be matched in M_H, and therefore destroyed in phase two. Since a_j is not matched to $\Pi^{-1}(a_j)$ or $\Pi(a_j)$, phase two must have decomposed and matched $P(a_j)\backslash\{a_j\}$, meaning that a_j was in M'_H at some point. If a_j remains matched in M'_H at the end of phase two, then a_i does not block with a_j – otherwise the loop cannot have terminated. If a_j is unmatched in M'_H at the end of phase two, then by part 2 of Lemma 9, a_i cannot find a_j acceptable, a contradiction. Hence, a_i does not block with a_j relative to M. □

We are now ready to present the main theorem of this section.

Theorem 11. *In polynomial-time, a matching M can be constructed with at most $(d-2)|\mathcal{P}_O^{ne}| + (d-1)|\mathcal{P}_O^{e}|$ blocking pairs, where $\mathcal{P}_O^{e}$ ($\mathcal{P}_O^{ne}$) is the set of elitist (respectively, non-elitist) odd parties in $\mathcal{P}$.*

Proof. The algorithm clearly runs in polynomial-time. To establish the upper bound, Lemma 5 and Corollary 6 imply that we need only prove an upper bound on the number of blocking pairs involving agents a_r in odd parties of size ≥ 3 that are not matched to $\Pi(a_r)$ or $\Pi^{-1}(a_r)$. There is exactly one such agent per odd party.

First, observe that by property (1) of the stable partition, a_r cannot block with $a_{r+1} = \Pi(a_r)$, who is matched to $a_{r+2} = \Pi(a_{r+1})$. Hence, a_r can conceivably block with all the other agents on his list, and is therefore in at most $d-1$ blocking pairs. This establishes the claimed number of blocking pairs for the elitist odd parties. Next, we show that we can identify another agent on a_r's list that he does not block with if his party was decomposed and matched in phases two or three.

Suppose $P(a_r)\backslash\{a_r\}$ was decomposed and matched during phase two. We consider two cases. First, suppose that a_r is matched in M at the end of phase two to an agent a_s. Since the pair $\{a_r, a_s\}$ is never removed from M, a_r cannot block with a_s or $\Pi(a_r)$, and is therefore in at most $d-2$ blocking pairs. Secondly, suppose that a_r is not matched in M at the end of phase two. For a_r to be unmatched, it must be that a_r was matched to some a_s in M'_H, and the pair $\{a_r, a_s\}$ was replaced with a different pair $\{a_s, b_t\}$ in the while loop of phase two, so that a_s prefers b_t to a_r. By part 3 of Lemma 9, the pair $\{a_s, b_t\}$ is never removed from M'_H, and is therefore added to M at the end of phase two, where it must remain. Hence, a_r does not block with a_s or $\Pi(a_r)$, and is thus in at most $d-2$ blocking pairs.

If $P(a_r)\backslash\{a_r\}$ was decomposed and matched in phase three, then a_r was chosen because there exists an $a_s \neq \Pi(a_r)$ that he prefers to $\Pi^{-1}(a_r)$. If a_s is not in his party, then by part 3 of Lemma 10, a_s does not block with a_r. If, instead a_s is in $P(a_r)$, then a_s is matched to either $\Pi(a_s)$ or $\Pi^{-1}(a_s)$ when $P(a_r)\backslash\{a_r\}$ is decomposed and matched in phase three, and does not block with a_r. Thus, a_r is in at most $d-2$ blocking pairs. □

5.4 The performance guarantee

We now show that the matching M implies an approximation algorithm with a performance guarantee of $2d-3$. Let M^* be an optimal solution, i.e., a matching such that $|bp(M^*)| = bp(I)$. Let $S = bp(M)$ and let $S^* = bp(M^*)$. For ease of notation, let $q_1 = |\mathcal{P}_O^{ne}|$ and $q_2 = |\mathcal{P}_O^{e}|$ with $q = q_1 + q_2$. The following lemma provides an additional lower bound (to that of Proposition 7) on the optimal solution in terms of the number of elitist odd parties in $\mathcal{P}$.

Lemma 12. *Let M^* be a matching with the minimum number of blocking pairs, and S^* the set of blocking pairs relative to M^*. Then, $|S^*| \geq |\mathcal{P}_O^e|$, where $\mathcal{P}_O^e$ is the set of elitist odd parties in G.*

Proof. Let $P = (u_0, \ldots, u_k)$ denote any arbitrary elitist odd party. Since P contains an odd number of agents, in any matching M for G, at least one agent $a_i \in P$ must be either unmatched, or matched to at most his third choice, a_j. Since agent $\Pi^{-1}(a_i)$ ranks a_i first on his list, $\{a_i, \Pi^{-1}(a_i)\}$ is a blocking pair for M. Therefore, M contains at least one blocking pair for each elitist odd party, and thus S^* contains at least $|\mathcal{P}_O^e|$ blocking pairs. □

The proof of the overall performance guarantee consists of two cases.

Case 1: $q_1 \leq q/2$. We have the following bounds (the third of which is due to Lemma 12): $q_1 \leq \frac{q}{2}$, $q_2 \geq \frac{q}{2}$, and $|S^*| \geq q_2$. By Theorem 11 it follows that:

$$|S| \leq (d-2)q_1 + (d-1)q_2 \leq \frac{(d-2)q}{2} + (d-1)|S^*|$$

Hence,

$$\frac{|S|}{|S^*|} \leq \frac{(d-2)q}{2|S^*|} + (d-1) \leq \frac{(d-2)q}{2}\left(\frac{2}{q}\right) + (d-1) \leq 2d-3.$$

Case 2: $q_1 > q/2$. We have the following bounds (the third of which is due to Proposition 7): $q_1 > \frac{q}{2}$, $q_2 < \frac{q}{2}$, and $|S^*| \geq \frac{q}{2}$. It follows by Theorem 11 that:

$$|S| \leq (d-2)q_1 + (d-1)q_2 = (d-2)q_1 + (d-1)(q-q_1) \leq \frac{(2d-3)q}{2}.$$

Hence,

$$\frac{|S|}{|S^*|} \leq \frac{(2d-3)q}{2|S^*|} \leq 2d-3.$$

This leads us to the following theorem.

Theorem 13. *For each fixed $d \geq 3$, MIN BP d-SRI is approximable within $2d-3$.*

Lastly, we note that in the absence of elitist odd parties, we have that the number of blocking pairs relative to M is at most $(d-2)q_1$, the number of odd parties. Since $\frac{q_1}{2}$ is, by Proposition 7, a lower bound on an optimal solution, we have the following theorem.

Theorem 14. *For each fixed $d \geq 3$, MIN BP d-SRI is approximable within $2d-4$ for instances where a stable partition contains no elitist odd party.*

6 Concluding remarks

As already mentioned in the Introduction, the performance guarantee of our approximation algorithm, as presented in Section 5, is in contrast with the findings of Hamada et al. [8], namely that the problem of finding a maximum matching with minimum number of blocking pairs is not approximable within $n^{1-\varepsilon}$, for any $\varepsilon > 0$ (unless P=NP), even if the underlying graph is bipartite and all preference lists are of length at most 3, where n is the number of agents. The intuition for this phenomenon is the following. When the overriding priority is to find a maximum matching, the number of blocking pairs in an optimal solution can increase in an uncontrolled way even if the instance admits a stable matching (which is always the case for a bipartite graph). However, in our case, we start with a stable partition and its structure, especially the number of odd parties of size at least three, provides both lower and upper limits for the minimum number of blocking pairs. This was a finding of Abraham et al. [1] and our new results can be seen as an extension of that theory. We suspect that the idea of using stable partitions for this and related problems can result in further and/or stronger findings.

An equivalent problem to MIN BP d-SRI is to find, given an SRI instance I, a smallest set of edges S such that $I \backslash S$ admits a stable matching, where $I \backslash S$ is the sub-instance of I obtained by deleting the acceptable pairs in S. Clearly the underlying graph of $I \backslash S$ is $G' = (A, E \backslash S)$, where $G = (A, E)$ is the underlying graph of I. In turn, this problem is polynomially equivalent to the following problem, which we call MAX STABLE SUBGRAPH: find a largest set of edges $S \subseteq E$ such that $I \backslash (E \backslash S)$ admits a stable matching (clearly $G' = (A, S)$ is the underlying graph of this sub-instance of I). Hence MAX STABLE SUBGRAPH is NP-hard in general.

Here is a simple 2-approximation for MAX STABLE SUBGRAPH, independent of the lengths of the preference lists. First, find a cut C in G having at least $m/2$ edges (such a cut exists, and is easily found in polynomial-time). Let E' be the set of cut edges. Then, $G' = (A, E')$ is bipartite and thus has a stable matching. Since m is an upper bound on any optimal solution, we have the claimed 2-approximation. Is there a better guarantee?

Note that the problem solved by Tan [16] (find a smallest set of agents that need to be removed from I in order to leave a stable matching) can be regarded as being polynomially equivalent to an induced subgraph counterpart of MAX STABLE SUBGRAPH, which is therefore solvable in polynomial time.

References

[1] D.J. Abraham, P. Biró, and D.F. Manlove. "Almost stable" matchings in the Roommates problem. In *Proceedings of WAOA '05: the 3rd Workshop on Approximation and Online Algorithms*, volume 3879 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2006.

[2] P. Berman, M. Karpinski, and Alexander D. Scott. Approximation hardness of short symmetric instances of MAX-3SAT. Electronic Colloquium on Computational Complexity Report, number 49, 2003.

[3] P. Biró, D.F. Manlove, and S. Mittal. Size versus stability in the marriage problem. *Theoretical Computer Science*, 411:1828–1841, 2010.

[4] K. Eriksson and O. Häggström. Instability of matchings in decentralized markets with various preference structures. *International Journal of Game Theory*, 36(3-4):409–420, 2008.

[5] P. Floréen, P. Kaski, V. Polishchuk, and J. Suomela. Almost stable matchings by truncating the Gale-Shapley algorithm. *Algorithmica*, 58(1):102–118, 2010.

[6] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.

[7] D. Gusfield and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.

[8] K. Hamada, K. Iwama, and S. Miyazaki. An improved approximation lower bound for finding almost stable stable maximum matchings. *Information Processing Letters*, 109(18):1036–1040, 2009.

[9] K. Hamada, K. Iwama, and S. Miyazaki. The hospitals/residents problem with quota lower bounds. In *Proceedings of ESA '11: the 19th European Symposium on Algorithms*, volume 6942 of *Lecture Notes in Computer Science*, pages 180–191. Springer, 2011.

[10] R.W. Irving. An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms*, 6:577–595, 1985.

[11] S. Khuller, S.G. Mitchell, and V.V. Vazirani. On-line algorithms for weighted bipartite matching and stable marriages. *Theoretical Computer Science*, 127:255–267, 1994.

[12] D.E. Knuth. *Mariages Stables.* Les Presses de L'Université de Montréal, 1976. English translation in *Stable Marriage and its Relation to Other Combinatorial Problems*, volume 10 of CRM Proceedings and Lecture Notes, American Mathematical Society, 1997.

[13] B.G. Pittel and R.W. Irving. An upper bound for the solvability probability of a random stable roommates instance. *Random Structures and Algorithms*, 5:465–486, 1994.

[14] A.E. Roth, T. Sönmez, and M. Utku Ünver. Pairwise kidney exchange. *Journal of Economic Theory*, 125:151–188, 2005.

[15] J.J.M. Tan. A necessary and sufficient condition for the existence of a complete stable matching. *Journal of Algorithms*, 12:154–178, 1991.

[16] J.J.M. Tan. Stable matchings and stable partitions. *International Journal of Computer Mathematics*, 39:11–20, 1991.