Noname manuscript No.
(will be inserted by the editor)

Polyhedral results for position based scheduling of chains on a single machine

Markó Horváth · Tamás Kis

Received: date / Accepted: date

Abstract We consider a scheduling problem where a set of unit-time jobs have to be sequenced on a single machine without any idle times between the jobs. Preemption of processing is not allowed. The processing cost of a job is determined by the position in the sequence, i.e., for each job and each position, there is an associated weight, and one has to determine a sequence of jobs which minimizes the total weight incurred by the positions of the jobs. In addition, the ordering of the jobs must satisfy the given chain precedence constraints. In this paper we investigate the polyhedron associated with a special case of the problem where each chain has length two. We show that optimizing over this polyhedron is strongly NP-hard, however, we present a class of facet-defining inequalities along with a polynomial-time separation procedure. We generalize these results to the case of chains with lengths at most two. Finally, we present our computational results that show that separating these inequalities can significantly improve a linear programming based branch-and-bound procedure to solve the problem.

Keywords scheduling · polyhedra · cutting planes

This work has been supported by the OTKA grant K112881, and by the grant GINOP-2.3.2-15-2016-00002 of the Ministry of National Economy of Hungary.

Markó Horváth
Institute for Computer Science and Control, Hungarian Academy of Sciences, H1111 Budapest, Kende str. 13–17, Hungary
ORCID: 0000-0002-1062-6272

Tamás Kis
Institute for Computer Science and Control, Hungarian Academy of Sciences, H1111 Budapest, Kende str. 13–17, Hungary
Tel.: +36-1-279-6156
Fax: +36-1-466-7503
E-mail: tamas.kis@sztaki.mta.hu
ORCID: 0000-0002-2759-1264

1 Introduction

We consider a scheduling problem where a set of unit-time jobs have to be sequenced on a single machine without any idle times between the jobs. Preemption of processing is not allowed. The ordering of the jobs must satisfy a given precedence relation derived from a directed acyclic graph. The processing cost of a job is determined by the position in the sequence, i.e., for each job and each position, there is an associated weight (which can be any rational number), and one has to determine a sequence of jobs which minimizes the total weight incurred by the positions of the jobs. In the following we consider the case of chain precedence constraints with chain-lengths at most two, that is, each job has at most one immediate predecessor and at most one immediate successor, and each chain consists of at most two jobs.

Formally, let $\mathcal{J} = \{J_1, \ldots, J_n\}$ be the set of unit-time jobs, that is, each job J_j has processing time $p_j = 1$. For a given schedule S and job J_j let $\sigma_j^S \in \{1, \ldots, n\}$ indicating the position of the job in the sequence (that is, $\sigma_j^S = k$ if exactly $k-1$ jobs are scheduled before J_j). For each job J_j and position k given a weight $w_{j,k} \in \mathbb{Q}$, thus the weight of job J_j for a given schedule S is w_{j,σ_j^S}. The goal of the problem is to determine a schedule S that minimizes the total weight $\sum_{j=1}^n w_{j,\sigma_j^S}$. Using the classification of deterministic sequencing and scheduling problems introduced by Graham et al. (1979) we denote the problem as $1\,|p_j = 1|\sum w_{j,\sigma_j}$. In the case of precedence relations we have a directed acyclic graph where the nodes correspond to the jobs, and if there is an arc between nodes i and j, then job J_i must be processed before job J_j. This problem is denoted as $1\,|prec, p_j = 1|\sum w_{j,\sigma_j}$, and if the directed acyclic graph decomposes into chains, then the problem is $1\,|chains, p_j = 1|\sum w_{j,\sigma_j}$. Note that problem $1\,|p_j = 1|\sum w_{j,\sigma_j}$ is equivalent to the well-known assignment problem (Kuhn, 1955), thus the problem $1\,|prec, p_j = 1|\sum w_{j,\sigma_j}$ can be considered as a generalized assignment problem where the set of positions is ordered, and the assignment must satisfy the given precedence constraints.

In this paper we investigate the polyhedra associated with special cases of the problem $1\,|chains, p_j = 1|\sum w_{j,\sigma_j}$. First, we consider the case where each chain has length two (i.e., the precedence graph is a directed perfect matching), that is, each node has either in-degree exactly one or out-degree exactly one. We denote this problem as $1\,|2\text{-}chains, p_j = 1|\sum w_{j,\sigma_j}$, where expression 2-*chains* indicates that each chain has length two, that is, consists of exactly two jobs. Then, we generalize the results of this problem to the case where the precedence constraints consist of chains with length at most two. The associated constraint is denoted by $chains, chain\text{-}length \in \{1, 2\}$.

The purpose of the paper at hand is to investigate the polyhedral structure of problems of the form $1\,|chains, p_j = 1|\,\gamma$. In Section 3 we give an integer programming formulation for the problem $1\,|prec, p_j = 1|\circ$ (i.e, we consider feasibility, but neglect the optimality criterion), then in Section 4.1 we simplify it to the problem $1\,|2\text{-}chains, p_j = 1|\circ$. In Section 4.2 we show that optimizing over the set of feasible solutions is NP-hard in the strong sense by showing that the problem $1\,|2\text{-}chains, p_j = 1|\sum w_{j,\sigma_j}$ is strongly NP-hard. Note that the latter result implies that the problem $1\,|prec, p_j = 1|\sum w_{j,\sigma_j}$ is NP-hard in the strong sense even in the case of chain precedence constraints with chain-lengths at most two. In Section 4.3 we determine the dimension of the polyhedron of the feasible solutions of $1\,|2\text{-}chains, p_j = 1|\,\circ$. In Section 4.4 we present a class of valid inequalities for the given formulation, we prove that these inequalities are facet-defining, and we show that they can be separated in polynomial time. In Section 5 we generalize these inequalities for the problem $1\,|chains, chain\text{-}length \in \{1,2\}, p_j = 1|\,\gamma$. Finally, in Section 6 we present our computational experiments, where we show that separating our inequalities can significantly improve a linear programming based branch-and-bound procedure to solve the problems $1\,|2\text{-}chains, p_j = 1|\sum w_{j,\sigma_j}$ and $1\,|chains, chain\text{-}length \in \{1,2\}, p_j = 1|\sum w_{j,\sigma_j}$.

2 Literature Review

Lenstra and Rinnooy Kan (1980) and Leung and Young (1990) present complexity results for scheduling unit-time jobs on a single machine with chain precedence constraints, i.e., problems of the form $1\,|chains, p_j = 1|\,\gamma$. Clearly, the problems with $\gamma = C_{\max}$ and $\gamma = \sum w_j C_j$ are polynomially solvable. Lenstra and Rinnooy Kan (1980) and Leung and Young (1990) show that problems with $\gamma = \sum U_j$ and $\gamma = \sum T_j$ are NP-hard, respectively. Our results in this paper imply that the problem with $\gamma = \sum w_{j,\sigma_j}$ is NP-hard in the strong sense even if each chain in the precedence relation has length 2.

Wan and Qi (2010) introduce new scheduling models where time slot costs have to be taken into consideration. In their models the planning horizon is divided into $K \geq \sum_{j=1}^{n} p_j$ time slots with unit length, where the kth time slot has cost π_k, and the time slot cost of a job J_j with starting time t is $\sum_{k\in s_j} \pi_k$, where $s_j = \{t+1, \ldots, t+p_j\}$. The objective of their models is a combination of the total time slot cost with a traditional scheduling criterion, that is, they consider problems of the form $1\,|slotcost|\,\gamma + \sum_j \sum_{k\in s_j} \pi_k$. Wan and Qi (2010) show that in case of non-decreasing time slot costs (that is, $\pi_1 \leq \ldots \leq \pi_K$) the problem can be reduced to one without slot costs. Under the assumption of arbitrarily varied time slot costs they prove that the problems with $\gamma = \sum C_j$, $\gamma = L_{\max}$, $\gamma = T_{\max}$, $\gamma = \sum U_j$ and $\gamma = \sum T_j$ are strongly NP-hard. They also show that in case of non-increasing time slot costs some of these problems can be solved in polynomial or pseudo-polynomial time. Zhao et al. (2016) prove that in case of non-increasing time slot costs, the problem $1\,|slotcost|\sum(C_j + \sum_{k\in s_j} \pi_k)$ is NP-hard in the strong sense. Kulkarni and Munagala (2012) introduce a model similar to that of (Wan and Qi, 2010), however, they deal with online algorithms to minimize the total time slot costs plus the total weighted completion time. Note that the problem investigated in this paper can be considered as a generalization of a special case of the model of Wan and Qi (2010). That is, in case of unit-time jobs (with $K = \sum_{j=1}^{n} p_j = n$) the problem $1\,|slotcost, p_j = 1|\sum_j \sum_{k\in s_j} \pi_k$ is similar to that of $1\,|p_j = 1|\sum w_{j,\sigma_j}$, however, in the latter problem the time slot costs depend on the jobs.

3 Problem formulation

Recall that $\mathcal{J} = \{J_1, \ldots, J_n\}$ is the set of unit-time jobs, and let $\mathcal{P} = \{1, \ldots, n\}$ be the set of positions. Let $D = (V, A)$ be the directed acyclic precedence graph with node set $V = \{1, \ldots, n\}$. The presence of a directed i–j path in D implies that job J_i have to be assigned to an earlier position than job J_j (that is, job J_i is a predecessor of job J_j, and job J_j is a successor of job J_i). Let $J_i \prec\!\prec J_j$ (or equivalently $i \prec\!\prec j$) denote if J_i is a predecessor of job J_j. In addition, let $J_i \prec J_j$ if job J_i is an immediate predecessor of job J_j, that is, $J_i \prec\!\prec J_j$, but there is no job J_k such that $J_i \prec\!\prec J_k \prec\!\prec J_j$.

Let $x_{i,j}$ be the binary variable indicating whether job J_i is assigned to position j. The problem $1\,|prec, p_j = 1|\circ$ can be formulated as

$$\sum_{j=1}^{n} x_{i,j} = 1,\ \ i \in \{1, \ldots, n\}, \tag{1}$$

$$\sum_{i=1}^{n} x_{i,j} = 1,\ \ j \in \{1, \ldots, n\}, \tag{2}$$

$$\sum_{j=1}^{k+1} x_{i_2,j} \le \sum_{j=1}^{k} x_{i_1,j},\ \ i_1 \prec i_2,\ k \in \{1, \ldots, n-1\}, \tag{3}$$

$$x_{i,j} \in \{0, 1\},\ \ i \in \{1, \ldots, n\},\ j \in \{1, \ldots, n\}, \tag{4}$$

where constraints (1) and (2) model the job-position assignment constraints, and constraint (3) ensures that the precedence constraints are satisfied. Let $\mathcal{J}_i^+ = \{J_{i'} \in \mathcal{J} : J_i \prec\!\prec J_{i'}\}$ ($\mathcal{J}_i^- = \{J_{i'} \in \mathcal{J} : J_{i'} \prec\!\prec J_i\}$) be the set of successors (predecessors) of job J_i. Clearly, for each feasible solution $x \in \{0, 1\}^{n \cdot n}$ of the problem (1)–(4) we have

$$x_{i,j} = 0,\ \ i \in \{1, \ldots, n\},\ j \in \{1, \ldots, |\mathcal{J}_i^-|\}, \tag{5}$$

$$x_{i,j} = 0,\ \ i \in \{1, \ldots, n\},\ j \in \{n - |\mathcal{J}_i^+| + 1, \ldots, n\}. \tag{6}$$

In case of problem $1\,|prec, p_j = 1|\sum w_{j,\sigma_j}$ for given weights $w : \mathcal{J} \times \mathcal{P} \to \mathbb{Q}$ the optimality criterion can be formulated as

$$\text{minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} x_{i,j}.$$

4 Problem $1\,|2\text{-}chains, p_j = 1|\,\gamma$

In this section we investigate the problem $1\,|2\text{-}chains, p_j = 1|\,\gamma$. Recall that in this problem the set of jobs has even cardinality, and each job has either exactly one predecessor or exactly one successor. In Section 4.1 we reformulate the problem described in the previous section, and define the set Q^{2C} of its feasible solutions. In Section 4.2 we show that optimizing over Q^{2C}, i.e., the problem $1\,|2\text{-}chains, p_j = 1|\sum w_{j,\sigma_j}$, is strongly NP-hard. In Section 4.3 we determine the dimension of the polytope Q^{2C}. In Section 4.4 we present a class of valid inequalities for Q^{2C} that are facet-defining, and we also provide a polynomial time algorithm for separating these inequalities. For basic concepts of polyhedral combinatorics we refer the reader to (Nemhauser and Wolsey, 1988).

4.1 Problem formulation

In order to simplify our notation, in this section let $\mathcal{J} = \{J_1, \ldots, J_{2n}\}$ be the set of unit-time jobs, and $\mathcal{C} = \{C_1, \ldots, C_n\}$ be the set of chain-precedence constraints, where $C_i = (J_{2i-1}, J_{2i})$, that is, $J_{2i-1} \prec J_{2i}$ for each $i \in \{1, \ldots, n\}$. We say that job J_{2i-1} (J_{2i}) is the first (second) job of chain C_i. In addition, let $\mathcal{P} = \{1, \ldots, 2n\}$ be the set of positions.

Let $s_{i,j}$ ($e_{i,j}$) indicate whether the first (second) job of chain $C_i \in \mathcal{C}$ is assigned to position $j \in \mathcal{P}$. Using these variables we reformulate (1)–(6) as:

$$\sum_{j=1}^{2n} s_{i,j} = 1, \;\; i \in \{1, \ldots, n\}, \tag{7}$$

$$\sum_{j=1}^{2n} e_{i,j} = 1, \;\; i \in \{1, \ldots, n\}, \tag{8}$$

$$s_{i,2n} = 0, \;\; i \in \{1, \ldots, n\}, \tag{9}$$

$$e_{i,1} = 0, \;\; i \in \{1, \ldots, n\}, \tag{10}$$

$$\sum_{i=1}^{n} s_{i,1} = 1, \tag{11}$$

$$\sum_{i=1}^{n} (s_{i,j} + e_{i,j}) = 1, \;\; j \in \{2, \ldots, 2n-1\}, \tag{12}$$

$$\sum_{i=1}^{n} e_{i,2n} = 1, \tag{13}$$

$$\sum_{j=1}^{k+1} e_{i,j} \leq \sum_{j=1}^{k} s_{i,j}, \;\; i \in \{1, \ldots, n\}, \; k \in \{1, \ldots, 2n-2\}. \tag{14}$$

Constraints (7)–(8) and (11)–(13) are the job-position assignment constraints (see (1) and (2)). Constraints (9)–(10) ensure that each first-job precedes the corresponding second-job (see (3)). Finally, constraints (9)–(10) forbid to assign a first-job to the last, or a second-job to the first position (see (5)–(6)).

Let $P_n^{2C} \subseteq \{0,1\}^{n\cdot(2n)} \times \{0,1\}^{n\cdot(2n)}$ be the set of incidence vectors corresponding to feasible job-position assignments, and let $Q_n^{2C} = conv(P_n^{2C})$. By construction, we have the following proposition.

Proposition 1 $P_n^{2C} = \{(s,e) \in \{0,1\}^{n\cdot 2n} \times \{0,1\}^{n\cdot 2n} : \; (s,e)$ *satisfies* $(7) - (14)\}$.

For a given point $P = (s,e) \in P_n^{2C}$, let $s(P,i) = j$ ($e(P,i) = j$) if $s_{i,j} = 1$ ($e_{i,j} = 1$). For a given $i \in \{1, \ldots, n\}$ let $\sigma_i(P)$ be a 2-dimensional vector such that $\sigma_i(P) = (s(P,i), e(P,i))$, and $\sigma(P)$ be a $2n$-dimensional vector such that $\sigma(P) = (\sigma_1(P), \ldots, \sigma_n(P))$. For example, for the point P indicated in Figure 1 we have $P = (1,0,0,0, \; 0,1,0,0, \; 0,0,1,0, \; 0,0,0,1)$, $\sigma_1(P) = (1,3)$, $\sigma_2(P) = (2,4)$, and $\sigma(P) = (1,3,2,4)$.

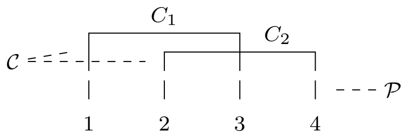

Fig. 1 Representation of point $P = (s, e) \in P_2^{2C}$ with $s_{1,1} = e_{1,3} = s_{2,2} = e_{2,4} = 1$

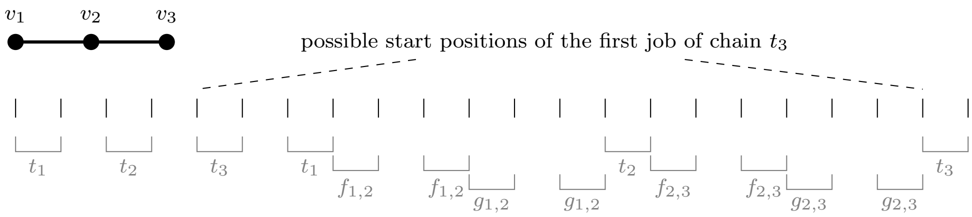

Fig. 2 Construction for the 2-length path

4.2 Optimizing over Q_n^{2C}

In Theorem 1 we will show that optimizing over the polyhedron Q_n^{2C} is NP-hard in the strong sense. We will transform the Independent Set (IS) problem to problem (15). An instance of IS is given by an undirected graph $G = (V, E)$ with node set $V = \{v_1, \ldots, v_n\}$, and a maximum size subset of nodes $I \subseteq V$ is sought such that for each edge $\{u, v\} \in E$, $|\{u, v\} \cap I| \leq 1$. The basic idea of the transformation can be seen in Figure 2, where we depict the construction for the 2-length path. Briefly stated, we will create a chain t_i for each node v_i and two chains $f_{i,j}$ and $g_{i,j}$ for each edge $\{v_i, v_j\}$ of the IS instance, and some additional dummy chains. To each of these chains we will designate two potential start and two potential end positions. On the one hand, by determining appropriate weights we will force that in each solution with non-negative total weight, each of these chains either starts/ends at its first start/end position, or at its second start/end position. In Figure 2 we depict the two potential states of these chains. On the other hand, by designating these positions properly, we will force that each solution with non-negative total weight represents an independent set in the IS instance and vice versa. Namely, a node is in the independent set if and only if the corresponding chain starts/ends its second start/end position. For example, in Figure 3 we depict the solution that represents the independent set $\{v_2\}$ (without the dummy chains). Note that since chain t_2 starts/ends at its second start/end position, i.e., v_2 is in the independent set, thus chains $g_{1,2}$, $f_{1,2}$ and therefore t_1 should start/end at its first start/end position, i.e., v_1 cannot be in the independent set. Similarly, t_3 cannot start/end at its second start/end position, that is, v_3 cannot be in the independent set. In Figure 4 we depict the solution that represents the independent set $\{v_1, v_3\}$ (without the dummy chains).

Theorem 1 *Let $w^s, w^e : \{1, \ldots, n\} \times \{1, \ldots, 2n\} \to \mathbb{R}$ be arbitrary weight functions. The problem*

$$\text{maximize} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{2n} w^s(i,j) s_{i,j} + \sum_{i=1}^{n} \sum_{j=1}^{2n} w^e(i,j) e_{i,j} : \ (s,e) \in P_n^{2C} \right\} \tag{15}$$

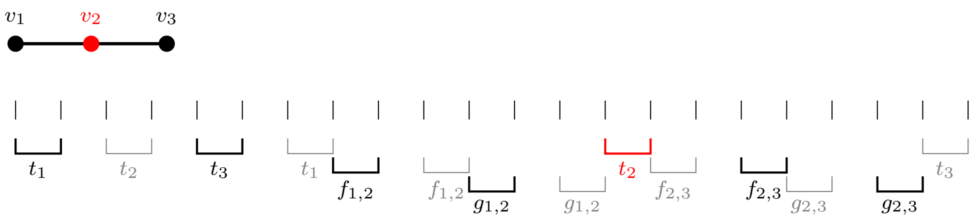

Fig. 3 Solution representing independent set $\{v_2\}$

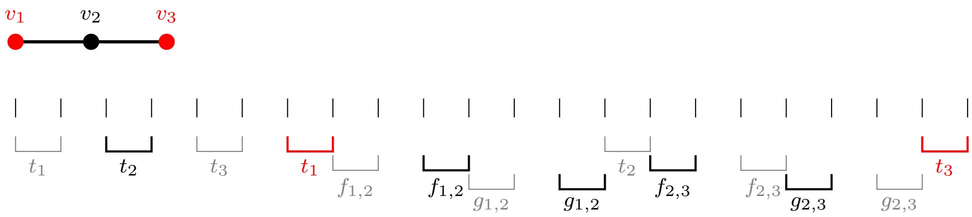

Fig. 4 Solution representing independent set $\{v_1, v_3\}$

is NP-hard in the strong sense.

Proof We transform the Independent Set (IS) problem to problem (15). Let $G = (V, E)$ be an instance for the independent set problem with node set $V = \{v_1, \ldots, v_n\}$, and let $\overrightarrow{E} = \{(v_i, v_j) : \{v_i, v_j\} \in E,\ i < j\}$ be the set of directed edges, i.e., we replace undirected edge $\{v_i, v_j\}$ by directed edge (v_i, v_j) for $i < j$. For a node v_i let $\operatorname{succ}(i) = \{v_j : (v_i, v_j) \in \overrightarrow{E}\}$ $(\operatorname{pred}(i) = \{v_j : (v_j, v_i) \in \overrightarrow{E}\})$ denote its immediate successors (predecessors).

For each $v_i \in V$ we create a node-chain t_i, and for each edge $(v_i, v_j) \in \overrightarrow{E}$ we create edge-chains $f_{i,j}$ and $g_{i,j}$. Let $\mathcal{T}_V = \{t_i : v_i \in V\}$ and $\mathcal{T}_{\overrightarrow{E}} = \{f_{i,j}, g_{i,j} : (v_i, v_j) \in \overrightarrow{E}\}$. To each node-chain $t_i \in \mathcal{T}_V$ we designate four distinct positions:

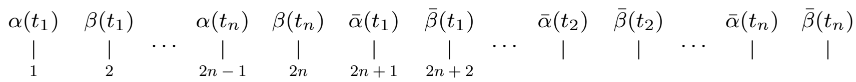

Fig. 5 Designated positions for node-chains

$\alpha(t_i) < \beta(t_i) < \bar{\alpha}(t_i) < \bar{\beta}(t_i)$ such that

i) $2i - 1 = \alpha(t_i) = \beta(t_i) - 1$, for all $i \in \{1, \ldots, n\}$,
ii) $2n + 1 = \bar{\alpha}(t_1) = \bar{\beta}(t_1) - 1$,
iii) $\bar{\beta}(t_i) < \bar{\alpha}(t_{i+1}) = \bar{\beta}(t_{i+1}) - 1$, for all $i \in \{1, \ldots, n-1\}$,

see Figure 5. To each edge-chain $f_{i,j} \in \mathcal{T}_{\overrightarrow{E}}$ we designate four distinct positions: $\alpha(f_{i,j}) < \beta(f_{i,j}) < \bar{\alpha}(f_{i,j}) < \bar{\beta}(f_{i,j})$. Consider a node $v_i \in V$ and its immediate successors $\operatorname{succ}(i) = \{v_{j_1}, \ldots, v_{j_{|\operatorname{succ}(i)|}}\}$. Let

iv) $\alpha(f_{i,j_1}) = \bar{\beta}(t_i)$,
v) $\alpha(f_{i,j_\ell}) = \beta(f_{i,j_\ell}) - 1 = \bar{\alpha}(f_{i,j_\ell}) - 2$, for all $\ell \in \{1, \ldots, |\operatorname{succ}(i)|\}$,

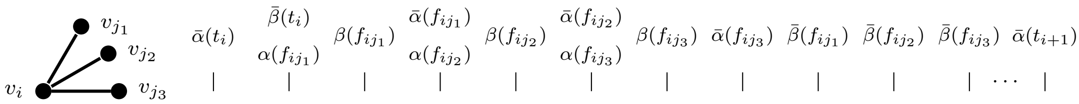

Fig. 6 Designated positions for edge-chains (part 1)

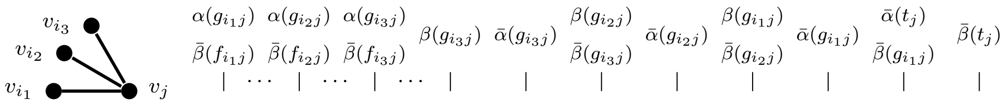

Fig. 7 Designated positions for edge-chains (part 2)

vi) $\bar\alpha(f_{i,j_\ell}) = \alpha(f_{i,j_{\ell+1}})$, for all $\ell \in \{1,\dots,|\text{succ}(i)|-1\}$,
vii) $\bar\alpha(f_{i,j_{|\text{succ}(i)|}}) = \bar\beta(f_{i,j_1}) - 1 = \bar\beta(f_{i,j_2}) - 2 = \ldots = \bar\beta(f_{i,j_{|\text{succ}(i)|}}) - |\text{succ}(i)|$,
viii) $\bar\beta(f_{i,j_{|\text{succ}(i)|}}) < \bar\alpha(t_{i+1})$,

see Figure 6. Finally, to each edge-chain $g_{i,j} \in \mathcal{T}_{\overrightarrow{E}}$ we designate four distinct positions: $\alpha(g_{i,j}) < \beta(g_{i,j}) < \bar\alpha(g_{i,j}) < \bar\beta(g_{i,j})$. Consider a node $v_j \in V$ and its immediate predecessors $\text{pred}(j) = \{v_{i_1},\dots,v_{i_{|\text{pred}(j)|}}\}$. Let

ix) $\bar\beta(g_{i_1,j}) = \bar\alpha(t_j)$,
x) $\beta(g_{i_\ell,j}) = \bar\alpha(g_{i_\ell,j}) - 1 = \bar\beta(g_{i_\ell,j}) - 2$, for all $\ell \in \{1,\dots,|\text{pred}(j)|\}$,
xi) $\beta(g_{i_\ell,j}) = \bar\beta(g_{i_{\ell+1},j})$, for all $\ell \in \{1,\dots,|\text{pred}(j)|-1\}$,
xii) $\alpha(g_{i_\ell,j}) = \bar\beta(f_{i_\ell,j})$, for all $\ell \in \{1,\dots,|\text{pred}(j)|\}$,
xiii) $\bar\beta(t_{j-1}) < \beta(g_{i_1,|\text{pred}(j)|})$,

see Figure 7.

For each $v_i \in V$ we have created 1 chain and designated 4 positions, and for each $(v_i,v_j) \in \overrightarrow{E}$ we have created 2 chains and designated 8 positions, however, positions $\alpha(f_{i,j})$, $\bar\beta(g_{i,j})$ and $\bar\beta(f_{i,j})$ coincide with other positions (see iv), vi), ix), xi), and xii)), hence we have $|V| + 2|E|$ chains, and $4|V| + 5|E|$ distinct positions. Thus, we also create $|V| + |E|$ dummy chains and $|E|$ dummy positions, therefore we have $2|V| + 3|E|$ chains and $2 \times (2|V| + 3|E|)$ positions, that is, we have a valid instance for problem (15).

Let $M > n$. For each $t_i \in \mathcal{T}_V$ let

$$w^s(t_i,j) = \begin{cases} -M \text{ if } j = \alpha(t_i), \\ 0 \text{ if } j = \bar\alpha(t_i), \\ -2M \text{ otherwise,} \end{cases} \quad \text{and} \quad w^e(t_i,j) = \begin{cases} M \text{ if } j = \beta(t_i), \\ 1 \text{ if } j = \bar\beta(t_i), \\ -2M \text{ otherwise.} \end{cases}$$

For each $t_{i,j} \in \mathcal{T}_{\overrightarrow{E}}$ ($t_{i,j}$ is either $f_{i,j}$ or $g_{i,j}$) let

$$w^s(t_{i,j},j) = \begin{cases} -M \text{ if } j = \alpha(t_{i,j}), \\ 0 \text{ if } j = \bar\alpha(t_{i,j}), \\ -2M \text{ otherwise,} \end{cases} \quad \text{and} \quad w^e(t_{i,j},j) = \begin{cases} M \text{ if } j = \beta(t_{i,j}), \\ 0 \text{ if } j = \bar\beta(t_{i,j}), \\ -2M \text{ otherwise.} \end{cases}$$

Finally, let $w^s(t,j) = w^e(t,j) = 0$, for each dummy chains t and for all $j = 1,\dots,(4|V| + 6|E|)$.

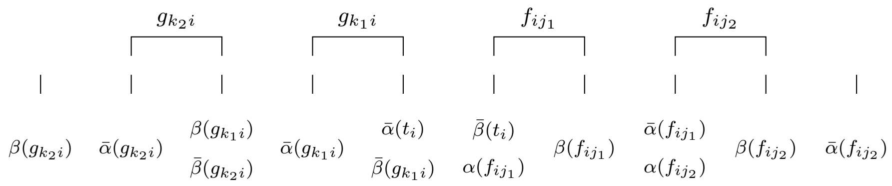

Fig. 8 Assignments for node $v_i \in V \setminus I$

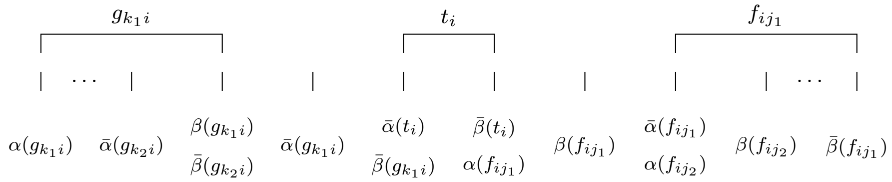

Fig. 9 Assignments for node $v_i \in I$

Remark 1 By construction, in any feasible solution for the constructed problem, for each $t \in \mathcal{T}_V$ we have

$$\sum_j w^s(t,j) + \sum_j w^e(t,j) = \begin{cases} 0 \text{ if } s_{t,\alpha(t)} = e_{t,\beta(t)} = 1, \\ 1 \text{ if } s_{t,\bar{\alpha}(t)} = e_{t,\bar{\beta}(t)} = 1, \\ \leq -M + 1 \text{ otherwise,} \end{cases}$$

and for each $t \in \mathcal{T}_{\overrightarrow{E}}$ we have

$$\sum_j w^s(t,j) + \sum_j w^e(t,j) = \begin{cases} 0 \text{ if } s_{t,\alpha(t)} = e_{t,\beta(t)} = 1 \text{ or } s_{t,\bar{\alpha}(t)} = e_{t,\bar{\beta}(t)} = 1, \\ \leq -M \text{ otherwise.} \end{cases}$$

Thus a solution for the created problem has non-negative total weight if and only if each chain $t \in \mathcal{T}_V \cup \mathcal{T}_{\overrightarrow{E}}$ starts/ends either its first start/end or its second start/end position.

Proposition 2 *Let $I \subseteq V$ an independent set. There is a solution for the constructed problem with total weight $|I|$.*

Proof If $v_i \notin I$, then let $s_{t_i,\alpha(t_i)} = e_{t_i,\beta(t_i)} = 1$, for each $(v_i, v_j) \in \overrightarrow{E}$ let $s_{f_{ij},\alpha(f_{ij})} = e_{f_{ij},\beta(f_{ij})} = 1$, and for each $(v_k, v_i) \in \overrightarrow{E}$ let $s_{g_{ki},\bar{\alpha}(g_{ki})} = e_{g_{ki},\bar{\beta}(g_{ki})} = 1$ (see Figure 8). Otherwise, if $v_i \in I$, then let $s_{t_i,\bar{\alpha}(t_i)} = e_{t_i,\bar{\beta}(t_i)} = 1$, for each $(v_i, v_j) \in \overrightarrow{E}$ let $s_{f_{ij},\bar{\alpha}(f_{ij})} = e_{f_{ij},\bar{\beta}(f_{ij})} = 1$, and for each $(v_k, v_i) \in \overrightarrow{E}$ let $s_{g_{ki},\alpha(g_{ki})} = e_{g_{ki},\beta(g_{ki})} = 1$ (see Figure 9). The variables for dummy chains can be arbitrary fixed.

On the one hand, we claim that this assignment yields a feasible solution. We need to show, that each position that designated to multiple jobs is assigned to a single job. It is easy to check that it is true for positions $\alpha(f_{i,j})$ and $\bar{\beta}(g_{i,j})$. We also know, that $\bar{\beta}(f_{i,j}) = \alpha(g_{i,j})$ for all edge $(v_i, v_j) \in \overrightarrow{E}$ (see xii)), however, we assigned position $\bar{\beta}(f_{i,j})$ to job $f_{i,j}$ and position $\alpha(g_{i,j})$ to job $g_{i,j}$ if and only if $v_i \in I$ and $v_j \in I$, respectively, however it is impossible, since I is independent. On the other hand, it is clear that the weight of the solution is equal to $|I|$. ◻

Proposition 3 *Consider a solution to the constructed problem with non-negative total weight W. There is an independent set $I \subseteq V$ with $|I| = W$.*

Proof Since W is non-negative, according to Remark 1, for each $t \in \mathcal{T}_V \cup \mathcal{T}_{\overrightarrow{E}}$ we have either $s_{t,\alpha(t)} = e_{t,\beta(t)} = 1$ or $s_{t,\bar{\alpha}(t)} = e_{t,\bar{\beta}(t)} = 1$. We claim that the node set $I = \{v_i \in V: \ s_{t_i,\bar{\alpha}(t_i)} = e_{t_i,\bar{\beta}(t_i)} = 1\}$ is independent.

Suppose for a contradiction that there is an edge $(v_i, v_j) \in \overrightarrow{E}$ such that $v_i, v_j \in I$. Let $succ(i) = \{v_{j_1}, \ldots, v_{j_{|succ(i)|}}\}$ be the set of the immediate successors of node v_i. Since $e_{t_i,\bar{\beta}(t_i)} = 1$ and by construction $\bar{\beta}(t_i) = \alpha(f_{ij_1})$, thus $s_{f_{ij_1},\alpha(f_{ij_1})} = 0$ and therefore $s_{f_{ij_1},\bar{\alpha}(f_{ij_1})} = e_{f_{ij_1},\bar{\beta}(f_{ij_1})} = 1$. Again, by construction $\bar{\alpha}(f_{ij_1}) = \alpha(f_{ij_2})$, thus $s_{f_{ij_2},\alpha(f_{ij_2})} = 0$ and therefore $s_{f_{ij_2},\bar{\alpha}(f_{ij_2})} = e_{f_{ij_2},\bar{\beta}(f_{ij_2})} = 1$. Similarly, we can show that $s_{f_{ij_\ell},\bar{\alpha}(f_{ij_\ell})} = e_{f_{ij_\ell},\bar{\beta}(f_{i,j_\ell})} = 1$ holds for all $\ell = 1, \ldots, |succ(i)|$, moreover, since $j = j_\ell$ for some $\ell \in \{1, \ldots, |succ(i)|\}$ we have $e_{f_{ij},\bar{\beta}(f_{ij})} = 1$.

Let $pred(j) = \{v_{i_1}, \ldots, v_{i_{|pred(j)|}}\}$ be the set of the immediate predecessors of node v_j. Similarly, we can show that $s_{g_{i_\ell j},\alpha(g_{i_\ell j})} = e_{g_{i_\ell j},\beta(g_{i_\ell j})} = 1$ holds for all $\ell = 1, \ldots, |pred(j)|$, and since $i = i_\ell$ for some $\ell \in \{1, \ldots, |pred(j)|\}$ we have $s_{g_{ij},\alpha(g_{ij})} = 1$.

To sum up, we have $e_{f_{ij},\bar{\beta}(f_{ij})} = s_{g_{ij},\alpha(g_{ij})} = 1$ which yields a contradiction, since by construction $\bar{\beta}(f_{ij}) = \alpha(g_{ij})$. □

Finally, it is easy to see that our transformation is a pseudo-polynomial transformation, thus the problem is NP-hard in the strong sense. □

Note that multiplying each weight value by -1 yields that the minimization version is also strongly NP-hard.

Corollary 1 *The problem $1\,|2\text{-}chains, p_j = 1|\sum w_{j,\sigma_j}$ is strongly NP-hard.*

Corollary 2 *The problem $1\,|chains, p_j = 1|\sum w_{j,\sigma_j}$ is strongly NP-hard even in the case of chains with length at most 2.*

Corollary 3 *The problem $1\,|prec, p_j = 1|\sum w_{j,\sigma_j}$ is strongly NP-hard.*

4.3 Dimension of Q_n^{2C}

In this section we investigate the dimension of the polyhedron Q_n^{2C}.

Theorem 2

$$\dim(Q_n^{2C}) = \begin{cases} 0 \text{ if } n = 1, \\ 5 \text{ if } n = 2, \\ 4n^2 - 6n + 1 \text{ if } n \geq 3. \end{cases}$$

Sketch of the proof of Theorem 2 ($n \geq 3$) We will show (see Theorem 3 and Proposition 6) that a minimal equation system for Q_n^{2C} has rank $6n-1$, thus the dimension of Q_n^{2C} is $4n^2 - (6n - 1)$.

Theorem 3 *Let $n \geq 3$. The equation system $\{(7)-(13)\}$ contains a minimal equation system for Q_n^{2C}.*

Proof We show that any equation which holds for all points of Q_n^{2C} is a linear combination of the equations (7)–(13). Assume that

$$\sum_{i=1}^{n}\sum_{j=1}^{2n}\alpha_{i,j}s_{i,j}+\sum_{i=1}^{n}\sum_{j=1}^{2n}\beta_{i,j}e_{i,j}=\gamma \tag{16}$$

holds for all $(s,e)\in Q_n^{2C}$. In order to show that equation (16) is a linear combination of equations (7)–(13) we explicitly create a linear combination (17), and in Propositions 4 and 5 we prove that (16) and (17) are the same. In these proposition we use Lemma 1, however, for its proofs we refer to the appendix.

Lemma 1 *For equation (16) the following statements hold:*

i) $\alpha_{p,j''}-\alpha_{p,j'}=\beta_{q,j''}-\beta_{q,j'}\quad\forall\, p,q\in\{1,\ldots,n\},\ 1<j'<j''<2n,$
ii) $\alpha_{p,j''}-\alpha_{p,j'}=\alpha_{q,j''}-\alpha_{q,j'}\quad\forall\, p,q\in\{1,\ldots,n\},\ 1\le j'<j''<2n,$
iii) $\beta_{p,j''}-\beta_{p,j'}=\beta_{q,j''}-\beta_{q,j'}\quad\forall\, p,q\in\{1,\ldots,n\},\ 1<j'<j''\le 2n.$

Note that in case of i) p may be equal to q.

Consider the linear combination of equations (7)–(13) with coefficients λ_i^7, λ_i^8, λ_i^9, λ_i^{10}, λ^{11}, λ_j^{12}, λ^{13} ($i\in\{1,\ldots,n\}$, $j\in\{2,\ldots,2n-1\}$), respectively, where

- $\lambda_i^7=\alpha_{i,1}-\alpha_{1,1}$ for all $i\in\{1,\ldots,n\}$,
- $\lambda_i^8=\beta_{i,2n}-\beta_{1,2n}+\beta_{1,2}-\alpha_{1,2}$ for all $i\in\{1,\ldots,n\}$,
- $\lambda_i^9=\alpha_{i,2n}-\alpha_{i,1}+\alpha_{1,1}$ for all $i\in\{1,\ldots,n\}$,
- $\lambda_i^{10}=\beta_{i,1}-\beta_{i,2n}+\beta_{1,2n}-\beta_{1,2}+\alpha_{1,2}$ for all $i\in\{1,\ldots,n\}$,
- $\lambda^{11}=\alpha_{1,1}$,
- $\lambda_j^{12}=\alpha_{1,j}$ for all $j\in\{2,\ldots,2n-1\}$,
- $\lambda^{13}=\beta_{1,2n}-\beta_{1,2}+\alpha_{1,2}$.

Let

$$\sum_{i=1}^{n}\sum_{j=1}^{2n}\hat\alpha_{i,j}s_{i,j}+\sum_{i=1}^{n}\sum_{j=1}^{2n}\hat\beta_{i,j}e_{i,j}=\hat\gamma \tag{17}$$

be the resulted equation. Note that the left-hand side can be written as

$$\sum_{i=1}^{n}\Big((\lambda_i^7+\lambda^{11})s_{i,1}+(\lambda_i^7+\lambda_i^9)s_{i,2n}+(\lambda_i^8+\lambda_i^{10})e_{i,1}+(\lambda_i^8+\lambda^{13})e_{i,2n}\Big)+$$
$$+\sum_{i=1}^{n}\sum_{j=2}^{2n-1}\Big((\lambda_i^7+\lambda_j^{12})s_{i,j}+(\lambda_i^8+\lambda_j^{12})e_{i,j}\Big).$$

Proposition 4 *For linear combination (17) the following statement holds:*

I) $\hat\alpha_{i,j}=\alpha_{i,j}$ *for all* $i\in\{1,\ldots,n\}$ *and* $j\in\{1,\ldots,2n\}$.

Proof Let $i\in\{1,\ldots,n\}$ be fixed. For $j=1$ we have

$$\hat\alpha_{i,1}=\lambda_i^7+\lambda^{11}=(\alpha_{i,1}-\alpha_{1,1})+\alpha_{1,1}=\alpha_{i,1},$$

and for $j=2n$ we have

$$\hat\alpha_{i,2n}=\lambda_i^7+\lambda_i^9=(\alpha_{i,1}-\alpha_{1,1})+(\alpha_{i,2n}-\alpha_{i,1}+\alpha_{1,1})=\alpha_{i,2n}.$$

For a given $j \in \{2, \ldots, 2n-1\}$ we have

$$\hat{\alpha}_{i,j} = \lambda_i^7 + \lambda_j^{12} = (\alpha_{i,1} - \alpha_{1,1}) + \alpha_{1,j} \overset{ii)}{=} \alpha_{i,j},$$

where for the last equation we use statement ii) of Lemma 1 with $p = 1$, $q = i$, $j' = 1$ and $j'' = j$. □

Proposition 5 *For linear combination (17) the following statement holds:*

II) $\hat{\beta}_{i,j} = \beta_{i,j}$ *for all* $i \in \{1, \ldots, n\}$ *and* $j \in \{1, \ldots, 2n\}$.

Proof Let $i \in \{1, \ldots, n\}$ be fixed. For $j = 1$ we have

$$\hat{\beta}_{i,1} = \lambda_i^8 + \lambda_i^{10} = (\beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2}) + (\beta_{i,1} - \beta_{i,2n} + \beta_{1,2n} - \beta_{1,2} + \alpha_{1,2}) = \beta_{i,1},$$

and for $j = 2n$ we have

$$\hat{\beta}_{1,2n} = \lambda_i^8 + \lambda^{13} = (\beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2}) + (\beta_{1,2n} - \beta_{1,2} + \alpha_{1,2}) = \beta_{i,2n}.$$

For a given $j \in \{2, \ldots, 2n-1\}$ we have

$$\hat{\beta}_{i,j} = \lambda_i^8 + \lambda_j^{12} = (\beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2}) + \alpha_{1,j} \overset{iii)}{=} \beta_{i,2} - \alpha_{1,2} + \alpha_{1,j} \overset{i)}{=} \beta_{i,j}.$$

since $\beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} = \beta_{i,2}$ according to statement iii) of Lemma 1, and $\beta_{i,2} - \alpha_{1,2} + \alpha_{1,j} = \beta_{i,j}$ due to statement i) of Lemma 1. □

Corollary 4 *Linear combination (17) yields equation (16).*

Proof According to Proposition 4 and 5, the left-hand sides of (17) and (16) are the same. Since both of them are satisfied by the points from the non-empty set P_n^{2C}, the right-hand sides also coincide. □

□

Proposition 6 *Let* $n \geq 3$. *The rank of the equation system* $\{(7) - (13)\}$ *is* $6n - 1$.

Proof Consider a linear combination of equations (7)–(13) with coefficients λ_i^7, λ_i^8, λ_i^9, λ_i^{10}, λ^{11}, λ_j^{12}, λ^{13} ($i \in \{1, \ldots, n\}$, $j \in \{2, \ldots, 2n-1\}$), respectively. This linear combination can be written as

$$\sum_{i=1}^{n} \Big((\lambda_i^7 + \lambda^{11}) s_{i,1} + (\lambda_i^7 + \lambda_i^9) s_{i,2n} + (\lambda_i^8 + \lambda_i^{10}) e_{i,1} + (\lambda_i^8 + \lambda^{13}) e_{i,2n} \Big) +$$

$$+ \sum_{i=1}^{n} \sum_{j=2}^{2n-1} \Big((\lambda_i^7 + \lambda_j^{12}) s_{i,j} + (\lambda_i^8 + \lambda_j^{12}) e_{i,j} \Big) = \lambda^{11} + \lambda^{13} + \sum_{i=1}^{n} \Big(\lambda_i^7 + \lambda_i^8 \Big) + \sum_{j=1}^{2n} \lambda_j^{12}.$$

It is the zero-equation (i.e., both of the right-hand side and the coefficients of the equation are zero) if and only if $\lambda_i^9 = -\lambda_i^7 = \lambda^{11}$, $\lambda_i^7 = -\lambda_j^{12} = \lambda_i^8$ and $\lambda_i^{10} = -\lambda_i^8 = \lambda^{13}$ hold for all $i \in \{1, \ldots, n\}$, $j \in \{2, \ldots, 2n-1\}$ and the right-hand side is zero. On the one hand, it is clear that we can easily choose non-zero coefficients that yield the zero-equation, thus the equations are linearly dependent. On the other hand, if we omit a single equation from (7)–(13), that is, we fix a single coefficient from $\lambda_i^7, \ldots, \lambda^{13}$ to zero, then all the remaining coefficients will be zero, that is, that remaining equations are linearly independent. Thus the equation system $\{(7) - (13)\}$ containing $6n$ equations has rank $6n - 1$. □

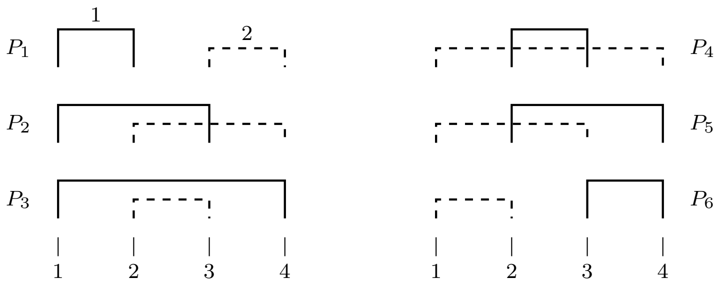

Fig. 10 The six points of P_2^{2C}

Proof (Theorem 2) In case of $n = 1$, P_1^{2C} consists of a single point P with $\sigma(P) = (1, 2)$, thus $\dim(Q_n^{2C}) = 0$.

In case of $n = 2$, $P_2^{2C} = \{P_1, \dots, P_6\}$, where $\sigma(P_1) = (1, 2, 3, 4)$, $\sigma(P_2) = (1, 3, 2, 4)$, $\sigma(P_3) = (1, 4, 2, 3)$, $\sigma(P_4) = (2, 3, 1, 4)$, $\sigma(P_5) = (2, 4, 1, 3)$, $\sigma(P_6) = (3, 4, 1, 2)$, see Figure 10. The linear combination of these points with coefficients $\lambda_1, \dots, \lambda_6$, respectively, is

$$(\lambda_1 + \lambda_2 + \lambda_3)s_{1,1} + (\lambda_4 + \lambda_5)s_{1,2} + \lambda_6 s_{1,3} + (\lambda_4 + \lambda_5 + \lambda_6)s_{2,1} + (\lambda_2 + \lambda_3)s_{2,2} + \\ + \lambda_1 s_{2,3} + \lambda_1 e_{1,2} + (\lambda_2 + \lambda_4)e_{1,3} + (\lambda_3 + \lambda_5 + \lambda_6)e_{1,4} + \lambda_6 e_{2,2} + \\ + (\lambda_3 + \lambda_5)e_{2,3} + (\lambda_1 + \lambda_2 + \lambda_4)e_{2,4}.$$

Clearly, we get the zero-vector if and only if $\lambda_1 = 0$, $\lambda_6 = 0$ and $\lambda_2 = -\lambda_3 = \lambda_5 = -\lambda_4$. On the one hand, we can easily choose non-zero $\lambda_2, \dots, \lambda_5$ coefficients to get the zero-vector, thus points $P_1, \dots, P_6$ are dependent. On the other hand, if we omit for example P_2, i.e., we fix $\lambda_2 = 0$, we could get the zero-vector if and only if $\lambda_1 = \ldots = \lambda_6 = 0$, that is, points P_1, P_3, P_4, P_5, P_6 are independent. Therefore $\dim(P_2^{2C}) = 5$.

Finally, assume that $n \geq 3$. Clearly, all points from Q_n^{2C} satisfy the equations (7)–(13). According to Theorem 3, (7)–(13) contains a minimal equation system for Q_n^{2C}, thus according to Proposition 6, the rank of such a minimal equation system is $6n - 1$. We have $4n^2$ variables, thus $\dim(Q_n^{2C}) = 4n^2 - (6n - 1)$. □

4.4 Parity inequalities

Theorem 4 *Let $n \geq 2$. The following inequalities are valid for Q_n^{2C}:*

$$\sum_{i \in S} \sum_{j=1}^{2k} (s_{i,j} - e_{i,j}) \leq |S| - 1 + \sum_{i \notin S} \sum_{j=1}^{2k} (s_{i,j} - e_{i,j}) \qquad \forall\ S \subseteq \{1, \dots, n\} : |S| \text{ is odd},\ k < n, \quad (18)$$

$$\sum_{i \in S} \sum_{j=1}^{2k-1} (s_{i,j} - e_{i,j}) \leq |S| - 1 + \sum_{i \notin S} \sum_{j=1}^{2k-1} (s_{i,j} - e_{i,j}) \qquad \forall\ S \subseteq \{1, \dots, n\} : |S| \text{ is even},\ k \leq n. \quad (19)$$

We say that a chain is active in interval $[j_1, j_2]$ (that is, between positions $j_1 < j_2$) if its first job is assigned to a position $j' \leq j_1$ and its second job is assigned to a position $j'' \geq j_2$. Let $z_{i,j} := \sum_{k=1}^{j}(s_{i,k} - e_{i,k})$ for all $i \in \{1,\ldots,n\}$ and $j \in \{1,\ldots,2n-1\}$. By this, $z_{i,j}$ is equal to 1, if and only if chain C_i is active in interval $[j, j+1]$ (otherwise it is zero). We can write (18) and (19) as

$$\sum_{i\in S} z_{i,2j} \leq |S| - 1 + \sum_{i\notin S} z_{i,2j} \qquad \forall\ S \subseteq \{1,\ldots,n\} : |S| \text{ is odd},\ j \in \{1,\ldots,n-1\}, \quad (20)$$

$$\sum_{i\in S} z_{i,2j-1} \leq |S| - 1 + \sum_{i\notin S} z_{i,2j-1} \qquad \forall\ S \subseteq \{1,\ldots,n\} : |S| \text{ is even},\ j \in \{1,\ldots,n\}, \quad (21)$$

or equivalently

$$1 \leq \sum_{i\in S}(1 - z_{i,2j}) + \sum_{i\notin S} z_{i,2j} \qquad \forall\ S \subseteq \{1,\ldots,n\} : |S| \text{ is odd},\ j \in \{1,\ldots,n-1\}, \quad (22)$$

$$1 \leq \sum_{i\in S}(1 - z_{i,2j-1}) + \sum_{i\notin S} z_{i,2j-1} \qquad \forall\ S \subseteq \{1,\ldots,n\} : |S| \text{ is even},\ j \in \{1,\ldots,n\}. \quad (23)$$

Proof (Theorem 4) Let (s, e) be an arbitrary point from P_n^{2C} and define z as above. Consider a set $S \subseteq \{1,\ldots,n\}$ with odd cardinality, and fix $j \in \{1,\ldots,n-1\}$. If $\sum_{i\in S} z_{i,2j} \leq |S| - 1$ then (20) clearly holds, thus assume that $\sum_{i\in S} z_{i,2j} = |S|$. Since the set $\{1,\ldots,2j\}$ of position has even cardinality, the number of chains that active in $[2j, 2j+1]$ (i.e, chains with first-job assigned to positions from $\{1,\ldots,2j\}$, and second-job assigned to position from $\{2j+1,\ldots,2n\}$) is also even, that is, $\sum_{i=1}^{n} z_{i,2j}$ is an even number. Therefore, since $\sum_{i\in S} z_{i,2j} = |S|$ is odd, there is at least one chain $i \notin S$ with $z_{i,2j} = 1$, thus (20) holds.

Similarly, one can prove that (21) (thus (19) and (23)) is valid for Q_n^{2C}. □

4.4.1 Separation

Theorem 5 *Inequalities (22) can be separated in polynomial time, that is, for a given vector $\bar{z} \in [0,1]^{n\cdot(2n-1)}$ and for a given index j the following problem can be solved in polynomial time:*

$$\text{maximize } \left\{ 1 - \left(\sum_{i\in S}(1 - \bar{z}_{i,2j}) + \sum_{i\notin S} \bar{z}_{i,2j} \right) :\ S \subseteq \{1,\ldots,n\},\ |S| \text{ is odd} \right\}.$$

Lemma 2 *Let $1 \geq v_1 \geq v_2 \geq \ldots \geq v_n \geq 0$, and let $f(S) := \sum_{i\in S}(1 - v_i) + \sum_{i\notin S} v_i$ for all $S \subseteq \{1,\ldots,n\}$. Consider the following problems:*

$$\text{minimize } \{f(S) :\ S \subseteq \{1,\ldots,n\},\ |S| \text{ is odd}\}, \quad (24)$$

$$\text{minimize } \{f(S) :\ S \subseteq \{1,\ldots,n\},\ |S| \text{ is even}\}. \quad (25)$$

a) *Let* $S_0 := \emptyset$ *and* $S_i := \{1, \ldots, i\}$ *for all* $i = 1, \ldots, n$. *There is an optimal solution* S_{OPT} *for problem (24) (problem (25)) such that* $S_{OPT} = S_k$ *for some* $k \in \{0, \ldots, n\}$.

b) *Let* $t := 0$ *if* $1-v_i > v_i$ *holds for all* $i = 1, \ldots, n$, *otherwise, let* $t := \max\{i : 1-v_i \leq v_i\}$. *One of the sets* S_{t-1}, S_t *and* S_{t+1} *is an optimal solution for problem (24) (problem (25)).*

Proof To prove statement a), consider an optimal solution S_{OPT} for problem (24) such that $p := \max\{i : S_i \subseteq S_{OPT}\}$ is maximal. Clearly, $p + 1 \notin S_{OPT}$. Suppose for a contradiction that there is an index $q > p + 1$ such that $q \in S_{OPT}$. Let $S' := (S_{OPT} \cup \{p + 1\}) \setminus \{q\}$. Now, we have $f(S_{OPT}) \leq f(S') = f(S_{OPT}) + (1 - v_{p+1}) - v_{p+1} - (1 - v_q) + v_q = f(S_{OPT}) + 2(v_q - v_{p+1}) \leq f(S_{OPT})$, thus S' is also an optimal solution for problem (24), however $p < \max\{i : S_i \subseteq S'\}$ which contradicts our assumption for S_{OPT}.

According to statement a) problems (24) and (25) can be restricted to subsets of the form S_i, $i = 0, \ldots, n$. For each $i < t$, $1 - v_{i+1} \leq v_{i+1}$, thus $f(S_{i+1}) = f(S_i) + (1 - v_{i+1}) - v_{i+1} \geq f(S_i)$. For each $i > t$, $1 - v_i > v_i$, thus $f(S_i) = f(S_{i-1}) + (1 - v_i) - v_i < f(S_{i+1})$. Therefore, we have

$$f(S_1) \geq \ldots \geq f(S_{t-1}) \geq f(S_t) \text{ and } f(S_t) < f(S_{t+1}) < \ldots < f(S_n),$$

thus if S_t has odd (even) cardinality, then it is an optimal solution for problem (24) (problem (25)), otherwise, $\arg\min\{f(S_{t-1}), f(S_{t+1})\}$ is an optimal solution for problem (24) (problem (25)). □

Proof (Theorem 5) For a given vector $\bar{z}$ and a given index j, let $v_i := \bar{z}_{i,2j}$, $1 \leq i \leq n$, and let $f(S) := \sum_{i \in S}(1 - v_i) + \sum_{i \notin S} v_i$ for all $S \subseteq \{1, \ldots, n\}$. Without loss of generality, we can assume that $v_1 \geq v_2 \geq \ldots \geq v_n$. By this, the separation problem is equivalent with problem (24) which can be solved in polynomial time according to Lemma 2. □

Similarly, one can prove the following theorem.

Theorem 6 *Inequalities (23) can be separated in polynomial time.*

4.4.2 Facets

In the section we show that some of the inequalities (18) are facet-defining. Similarly, one can show that a subset of inequalities (19) are also facet-defining.

Let $3 \leq t < n$ be a fixed odd number; $1 \leq k < n$ such that $t < 2k$ and $t < 2(n-k)$ hold; and $S \subseteq \{1, \ldots, n\}$ with cardinality t. To simplify our notation, without loss of generality, we assume that $S = \{1, \ldots, t\}$. The corresponding parity inequality is:

$$\sum_{i=1}^{t} \sum_{j=1}^{2k} (s_{i,j} - e_{i,j}) \leq t - 1 + \sum_{i=t+1}^{n} \sum_{j=1}^{2k} (s_{i,j} - e_{i,j}). \tag{26}$$

Remark 2 A point from P_n^{2C} satisfies (26) with equation if and only if

- exactly $t - 1$ chains from $\{1, \ldots, t\}$ and exactly 0 chain from $\{t + 1, \ldots, n\}$ are active in interval $[2k, 2k + 1]$, or
- exactly t chains from $\{1, \ldots, t\}$ and exactly 1 chain from $\{t+1, \ldots, n\}$ are active in interval $[2k, 2k + 1]$.

Sketch of the proof of the theorem Let $P_n^{EVEN} = \{(s,e) \in P_n^{2C} \ : (s,e) \text{ satisfies } (26)\}$ and $Q_n^{EVEN} = conv(P_n^{EVEN})$. To prove that inequalities (26) are facet-defining for Q_n^{2C} we will show that $\dim(Q_n^{EVEN}) = \dim(Q_n^{2C}) - 1$. To do this, we apply a similar procedure as in Section 4.3, that is, we will prove that equation system $\bar{Q}_n^{EVEN} := \{(7)-(13),(27)\}$ contains a minimal equation system for Q_n^{EVEN}, where we have

$$\sum_{i=1}^{t}\sum_{j=1}^{2k}(s_{i,j}-e_{i,j}) + \sum_{i=t+1}^{n}\sum_{j=1}^{2k}(e_{i,j}-s_{i,j}) = t-1. \tag{27}$$

On the one hand, according to Theorem 3, $rank(\bar{Q}_n^{EVEN}) \le rank(\{(7)-(13)\})+1$, thus $\dim(Q_n^{EVEN}) \ge \dim(Q_n^{2C}) - 1$. On the other hand, $Q_n^{EVEN} \subsetneq Q_n^{2C}$.

Theorem 7 *The equation system* $\bar{Q}_n^{EVEN} = \{(7)-(13),(27)\}$ *contains a minimal equation system for* Q_n^{EVEN}.

Proof Assume that

$$\sum_{i=1}^{n}\sum_{j=1}^{2n}\alpha_{i,j}s_{i,j} + \sum_{i=1}^{n}\sum_{j=1}^{2n}\beta_{i,j}e_{i,j} = \gamma \tag{28}$$

holds for all $(s,e) \in Q_n^{EVEN}$. In order to show that equation (28) is a linear combination of equations (7)–(13) and (27) we explicitly create a linear combination (29), and in Propositions 7–10 we prove that (28) and (29) are the same. In these proposition we use Lemmas 3 and 4, however for their proofs we refer to the appendix.

Lemma 3 *For equation (28) the following statements hold:*

i) $\alpha_{p,j''} - \alpha_{p,j'} = \alpha_{q,j''} - \alpha_{q,j'} \quad \forall\, p,q \in \{1,\dots,t\},\ 1 \le j' < j'' \le 2k,$
ii) $\alpha_{p,j''} - \alpha_{p,j'} = \alpha_{q,j''} - \alpha_{q,j'} \quad \forall\, p,q \in \{1,\dots,t\},\ 1 \le j' \le 2k < j'' \le 2n-1,$
iii) $\beta_{p,j''} - \beta_{p,j'} = \beta_{q,j''} - \beta_{q,j'} \quad \forall\, p,q \in \{1,\dots,t\},\ 2k < j' < j'' \le 2n,$
iv) $\beta_{p,j''} - \beta_{p,j'} = \beta_{q,j''} - \beta_{q,j'} \quad \forall\, p,q \in \{1,\dots,t\},\ 2 \le j' \le 2k < j'' \le 2n,$
v) $\alpha_{p,j''} - \alpha_{p,j'} = \beta_{q,j''} - \beta_{q,j'} \quad \forall\, p,q \in \{1,\dots,t\},\ 1 < j' < j'' \le 2k,$
vi) $\alpha_{p,j''} - \alpha_{p,j'} = \beta_{q,j''} - \beta_{q,j'} \quad \forall\, p,q \in \{1,\dots,t\},\ 2k < j' < j'' < 2n.$

Note that in case of v) and vi) p may be equal to q.

Lemma 4 *For equation (28) the following statements hold:*

vii) $\alpha_{p,j''} - \alpha_{p,j'} = \alpha_{\bar{q},j''} - \alpha_{\bar{q},j'} \quad \forall\, p \in \{1,\dots,t\},\ \bar{q} \in \{t+1,\dots,n\},\ 1 \le j' < j'' \le 2k,$
viii) $\beta_{p,j''} - \beta_{p,j'} = \beta_{\bar{q},j''} - \beta_{\bar{q},j'} \quad \forall\, p \in \{1,\dots,t\},\ \bar{q} \in \{t+1,\dots,n\},\ 2k < j' < j'' \le 2n,$
ix) $\alpha_{p,j''} - \alpha_{p,j'} = \beta_{\bar{q},j''} - \beta_{\bar{q},j'} \quad \forall\, p \in \{1,\dots,t\},\ \bar{q} \in \{t+1,\dots,n\},\ 1 < j' < j'' \le 2k,$
x) $\alpha_{p,j''} - \alpha_{p,j'} = \beta_{\bar{q},j''} - \beta_{\bar{q},j'} \quad \forall\, p \in \{1,\dots,t\},\ \bar{q} \in \{t+1,\dots,n\},\ 1 < j' \le 2k < j'' < 2n,$
xi) $\beta_{p,j''} - \beta_{p,j'} = \alpha_{\bar{q},j''} - \alpha_{\bar{q},j'} \quad \forall\, p \in \{1,\dots,t\},\ \bar{q} \in \{t+1,\dots,n\},\ 1 < j' \le 2k < j'' < 2n,$
xii) $\beta_{p,j''} - \beta_{p,j'} = \alpha_{\bar{q},j''} - \alpha_{\bar{q},j'} \quad \forall\, p \in \{1,\dots,t\},\ \bar{q} \in \{t+1,\dots,n\},\ 2k < j' < j'' < 2n.$

Consider the linear combination of equations (7)–(13) and (27) with coefficients λ_i^7, λ_i^8, λ_i^9, λ_i^{10}, λ^{11}, λ_j^{12}, λ^{13} and λ^{27}, $(i \in \{1,\dots,n\},\ j \in \{2,\dots,2n-1\})$ respectively, where

- $\lambda^{27} = \lambda$, where $\lambda := (\alpha_{1,2} - \alpha_{1,2k+1} - \beta_{1,2} + \beta_{1,2k+1})/2$,
- $\lambda_i^7 = \begin{cases} \alpha_{i,1} - \alpha_{1,1} & \text{if } i \in \{1,\dots,t\}, \\ \alpha_{i,1} - \alpha_{1,1} + 2\lambda & \text{if } i \in \{t+1,\dots,n\}, \end{cases}$
- $\lambda_i^8 = \mu_i$, where $\mu_i := \begin{cases} \beta_{1,2} - \alpha_{1,2} + 2\lambda & \text{if } i = 1, \\ \beta_{i,2n} - \beta_{1,2n} + \mu_1 & \text{if } i \in \{2,\dots,n\}, \end{cases}$
- $\lambda_i^9 = \begin{cases} \alpha_{1,2n} & \text{if } i = 1, \\ \alpha_{i,2n} - \alpha_{i,1} + \alpha_{1,1} & \text{if } i \in \{2,\dots,t\}, \\ \alpha_{i,2n} - \alpha_{i,1} + \alpha_{1,1} - 2\lambda & \text{if } i \in \{t+1,\dots,n\}, \end{cases}$
- $\lambda_i^{10} = \begin{cases} \beta_{i,1} + \lambda - \mu_i & \text{if } i \in \{1,\dots,t\}, \\ \beta_{i,1} - \lambda - \mu_i & \text{if } i \in \{t+1,\dots,n\}, \end{cases}$
- $\lambda^{11} = \alpha_{1,1} - \lambda$,
- $\lambda_j^{12} = \begin{cases} \alpha_{1,j} - \lambda & \text{if } j \in \{2,\dots,2k\}, \\ \alpha_{1,j} & \text{if } j \in \{2k+1,\dots,2n-1\}, \end{cases}$
- $\lambda^{13} = \beta_{1,2n} - \mu_1$.

Let

$$\sum_{i=1}^{n}\sum_{j=1}^{2n} \hat{\alpha}_{i,j} s_{i,j} + \sum_{i=1}^{n}\sum_{j=1}^{2n} \hat{\beta}_{i,j} e_{i,j} = \hat{\gamma} \tag{29}$$

be the resulting equation. Note that the left-hand side can be written as

$$\sum_{i=1}^{t}\Big((\lambda_i^7 + \lambda^{11} + \lambda^{27}) s_{i,1} + (\lambda_i^8 + \lambda_i^{10} - \lambda^{27}) e_{i,1}\Big) + \sum_{i=t+1}^{n}\Big((\lambda_i^7 + \lambda^{11} - \lambda^{27}) s_{i,1} + (\lambda_i^8 + \lambda_i^{10} + \lambda^{27}) e_{i,1}\Big) +$$
$$+ \sum_{i=1}^{n}\Big((\lambda_i^7 + \lambda_i^9) s_{i,2n} + (\lambda_i^8 + \lambda^{13}) e_{i,2n}\Big) + \sum_{i=1}^{t}\sum_{j=2}^{2k}\Big((\lambda_i^7 + \lambda_j^{12} + \lambda^{27}) s_{i,j} + (\lambda_i^8 + \lambda_j^{12} - \lambda^{27}) e_{i,j}\Big) +$$
$$+ \sum_{i=t+1}^{n}\sum_{j=2}^{2k}\Big((\lambda_i^7 + \lambda_j^{12} - \lambda^{27}) s_{i,j} + (\lambda_i^8 + \lambda_j^{12} + \lambda^{27}) e_{i,j}\Big) + \sum_{i=1}^{n}\sum_{j=2k+1}^{2n-1}\Big((\lambda_i^7 + \lambda_j^{12}) s_{i,j} + (\lambda_i^8 + \lambda_j^{12}) e_{i,j}\Big).$$

Proposition 7 *For linear combination (29) the following statement holds:*

I) $\hat{\alpha}_{i,j} = \alpha_{i,j}$ *for all* $i \in \{1,\dots,t\}$ *and* $j \in \{1,\dots,2n\}$.

Proof By construction, the statement clearly holds for $i = 1$. Let $i \in \{2,\dots,t\}$ be fixed. For $j = 1$ we have

$$\hat{\alpha}_{i,1} = \lambda_i^7 + \lambda^{11} + \lambda^{27} = (\alpha_{i,1} - \alpha_{1,1}) + (\alpha_{1,1} - \lambda) + \lambda = \alpha_{i,1},$$

and for $j = 2n$ we have

$$\hat{\alpha}_{i,2n} = \lambda_i^7 + \lambda_i^9 = (\alpha_{i,1} - \alpha_{1,1}) + (\alpha_{i,2n} + \alpha_{1,1} - \alpha_{i,1}) = \alpha_{i,2n}.$$

For a given $j \in \{2,\dots,2k\}$ we have

$$\hat{\alpha}_{i,j} = \lambda_i^7 + \lambda_j^{12} + \lambda^{27} = (\alpha_{i,1} - \alpha_{1,1}) + (\alpha_{1,j} - \lambda) + \lambda = \alpha_{1,j} - \alpha_{1,1} + \alpha_{i,1} \overset{i)}{=} \alpha_{i,j},$$

where for the last equation we use statement i) of Lemma 3 with $p = 1$, $q = i$, $j' = 1$, and $j'' = j$. Finally, for a given $j \in \{2k+1, \ldots, 2n-1\}$ we have

$$\hat{\alpha}_{i,j} = \lambda_i^7 + \lambda_j^{12} = \alpha_{1,j} - \alpha_{1,1} + \alpha_{i,1} \overset{ii)}{=} \alpha_{i,j},$$

where for the last equation we use statement ii) of Lemma 3 with $p = 1$, $q = i$, $j' = 1$ and $j'' = j$. □

Proposition 8 *For linear combination (29) the following statement holds:*

II) $\hat{\beta}_{i,j} = \beta_{i,j}$ *for all* $i \in \{1, \ldots, t\}$ *and* $j \in \{1, \ldots, 2n\}$.

Proof First, assume that $i = 1$. For $j = 1$ we have

$$\hat{\beta}_{1,1} = \lambda_1^8 + \lambda_1^{10} - \lambda^{27} = \mu_1 + (\beta_{1,1} + \lambda - \mu_1) - \lambda = \beta_{1,1},$$

and for $j = 2n$ we have

$$\hat{\beta}_{1,2n} = \lambda_1^8 + \lambda^{13} = \mu_1 + (\beta_{1,2n} - \mu_1) = \beta_{1,2n}.$$

For a given $j \in \{2, \ldots, 2k\}$ we have

$$\hat{\beta}_{1,j} = \lambda_1^8 + \lambda_j^{12} - \lambda^{27} = (\beta_{1,2} - \alpha_{1,2} + 2\lambda) + (\alpha_{1,j} - \lambda) - \lambda = \beta_{1,2} + \alpha_{1,j} - \alpha_{1,2} \overset{v)}{=} \beta_{1,j},$$

where the last equation clearly holds for $j = 2$, and for $2 < j$ we can use statement v) of Lemma 3 with $p = q = 1$, $j' = 2$ and $j'' = j$. For a given $j \in \{2k+1, \ldots, 2n-1\}$ we have

$$\hat{\beta}_{1,j} = \lambda_1^8 + \lambda_j^{12} = \beta_{1,2} + \alpha_{1,j} - \alpha_{1,2} + 2\lambda = \alpha_{1,j} - \alpha_{1,2k+1} + \beta_{1,2k+1} \overset{vi)}{=} \beta_{i,j},$$

according to statement vi) of Lemma 3 with $p = q = 1$, $j' = 2k+1$ and $j'' = j$. Now, let $i \in \{2, \ldots, t\}$. For $j = 1$ we have

$$\hat{\beta}_{i,1} = \lambda_i^8 + \lambda_i^{10} - \lambda^{27} = \mu_i + (\beta_{i,1} + \lambda - \mu_i) - \lambda = \beta_{i,1},$$

and for $j = 2n$ we have

$$\hat{\beta}_{1,2n} = \lambda_i^8 + \lambda^{13} = (\beta_{i,2n} - \beta_{1,2n} + \mu_1) + (\beta_{1,2n} - \mu_1) = \beta_{i,2n}.$$

For a given $j \in \{2, \ldots, 2k\}$ we have

$$\begin{aligned}\hat{\beta}_{i,j} &= \lambda_i^8 + \lambda_j^{12} - \lambda^{27} = (\beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2} + 2\lambda) + (\alpha_{1,j} - \lambda) - \lambda \\ &= \beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2} + \alpha_{1,j} \overset{v)}{=} \beta_{i,2n} - \beta_{1,2n} + \beta_{1,j} \overset{iv)}{=} \beta_{i,j},\end{aligned}$$

since $\beta_{1,2} - \alpha_{1,2} + \alpha_{1,j} = \beta_{1,j}$ according to statement v) of Lemma 3 with $p = q = 1$, $j' = 2$ and $j'' = j$, and $\beta_{i,2n} - \beta_{1,2n} + \beta_{1,j} = \beta_{i,j}$ due to statement iv) of Lemma 3 with $p = 1$, $q = i$, $j' = j$ and $j'' = 2n$. Finally, for a given $j \in \{2k+1, \ldots, 2n-1\}$ we have

$$\hat{\beta}_{i,j} = \lambda_i^8 + \lambda_j^{12} = \beta_{i,2n} - \beta_{1,2n} + \alpha_{1,j} - \alpha_{1,2k+1} + \beta_{1,2k+1} \overset{vi)}{=} \beta_{i,2n} - \beta_{1,2n} + \beta_{1,j} \overset{iv)}{=} \beta_{i,j},$$

since $\alpha_{1,j} - \alpha_{1,2k+1} + \beta_{1,2k+1} = \beta_{1,j}$ according to statement vi) of Lemma 3 with $p = q = 1$, $j' = 2k+1$ and $j'' = j$, and $\beta_{i,2n} - \beta_{1,2n} + \beta_{1,j} = \beta_{i,j}$ due to statement iv) of Lemma 3 with $p = 1$, $q = i$, $j' = j$ and $j'' = 2n$. □

Proposition 9 *For linear combination (29) the following statement holds:*
III) $\hat{\alpha}_{i,j} = \alpha_{i,j}$ *for all* $i \in \{t+1, \dots, n\}$ *and* $j \in \{1, \dots, 2n\}$.

Proof Let $i \in \{t+1, \dots, n\}$ be fixed. For $j = 1$ we have

$$\hat{\alpha}_{i,1} = \lambda_i^7 + \lambda^{11} - \lambda^{27} = (\alpha_{i,1} - \alpha_{1,1} + 2\lambda) + (\alpha_{1,1} - \lambda) - \lambda = \alpha_{i,1},$$

and for $j = 2n$ we have

$$\hat{\alpha}_{i,2n} = \lambda_i^7 + \lambda_i^9 = (\alpha_{i,1} - \alpha_{1,1} + 2\lambda) + (\alpha_{i,2n} - \alpha_{i,1} + \alpha_{1,1} - 2\lambda) = \alpha_{i,2n}.$$

For a given $j \in \{2, \dots, 2k\}$ we have

$$\hat{\alpha}_{i,j} = \lambda_i^7 + \lambda_j^{12} - \lambda^{27} = (\alpha_{i,1} - \alpha_{1,1} + 2\lambda) + (\alpha_{1,j} - \lambda) - \lambda = \alpha_{1,j} - \alpha_{1,1} + \alpha_{i,1} \overset{vii)}{=} \alpha_{i,j},$$

where for the last equation we use statement vii) of Lemma 4 with $p = 1$, $\bar{q} = i$, $j' = 1$ and $j'' = j$. Finally, for a given $j \in \{2k+1, \dots, 2n-1\}$ we have

$$\begin{aligned}\hat{\alpha}_{i,j} &= \lambda_i^7 + \lambda_j^{12} = (\alpha_{i,1} - \alpha_{1,1} + \alpha_{1,2} - \beta_{1,2} + \beta_{1,2k+1} - \alpha_{1,2k+1}) + \alpha_{1,j} \\ &\overset{vi)}{=} \alpha_{i,1} - \alpha_{1,1} + \alpha_{1,2} - \beta_{1,2} + \beta_{1,j} \overset{vii)}{=} \alpha_{i,2} - \beta_{1,2} + \beta_{1,j} \overset{xi)}{=} \alpha_{i,j},\end{aligned}$$

since $\beta_{1,2k+1} - \alpha_{1,2k+1} + \alpha_{1,j} = \beta_{1,j}$ according to statement vi) of Lemma 3 with $p = q = 1$, $j' = 2k+1$ and $j'' = j$, and $\alpha_{i,1} - \alpha_{1,1} + \alpha_{1,2} = \alpha_{i,2}$ according to statement vii) of Lemma 4 with $p = 1$, $\bar{q} = i$, $j' = 1$ and $j'' = 2$, and $\alpha_{i,2} - \beta_{1,2} + \beta_{1,j} = \alpha_{i,j}$ due to statement xi) of Lemma 4 with $p = 1$, $\bar{q} = i$, $j' = 2$ and $j'' = j$. □

Proposition 10 *For linear combination (29) the following statement holds:*
IV) $\hat{\beta}_{i,j} = \beta_{i,j}$ *for all* $i \in \{t+1, \dots, n\}$ *and* $j \in \{1, \dots, 2n\}$.

Proof Let $i \in \{t+1, \dots, n\}$ be fixed. For $j = 1$ we have

$$\hat{\beta}_{i,1} = \lambda_i^8 + \lambda_i^{10} + \lambda^{27} = \mu_i + (\beta_{i,1} - \lambda - \mu_i) + \lambda = \beta_{i,1},$$

and for $j = 2n$ we have

$$\hat{\beta}_{1,2n} = \lambda_i^8 + \lambda^{13} = (\beta_{i,2n} - \beta_{1,2n} + \mu_1) + (\beta_{1,2n} - \mu_1) = \beta_{i,2n}.$$

For a given $j \in \{2, \dots, 2k\}$ we have

$$\begin{aligned}\hat{\beta}_{i,j} = \lambda_i^8 + \lambda_j^{12} + \lambda^{27} &= \beta_{i,2n} + \alpha_{1,j} - \beta_{1,2n} - \alpha_{1,2k+1} + \beta_{1,2k+1} \\ &\overset{viii)}{=} \alpha_{1,j} - \alpha_{1,2k+1} + \beta_{i,2k+1} \overset{x)}{=} \beta_{i,j},\end{aligned}$$

since $\beta_{i,2n} - \beta_{1,2n} + \beta_{1,2k+1} = \beta_{i,2k+1}$ according to statement viii) of Lemma 4 with $p = 1$, $\bar{q} = i$, $j' = 2k+1$ and $j'' = 2n$, and $\alpha_{1,j} - \alpha_{1,2k+1} + \beta_{i,2k+1} = \beta_{i,j}$ due to statement x) of Lemma 4 with $p = 1$, $\bar{q} = i$, $j' = j$ and $j'' = 2k+1$. Finally, for a given $j \in \{2k+1, \dots, 2n-1\}$ we have

$$\hat{\beta}_{i,j} = \lambda_i^8 + \lambda_j^{12} = \beta_{i,2n} - \beta_{1,2n} + \alpha_{1,j} - \alpha_{1,2k+1} + \beta_{1,2k+1} \overset{vi)}{=} \beta_{i,2n} - \beta_{1,2n} + \beta_{1,j} \overset{viii)}{=} \beta_{i,j},$$

since $\alpha_{1,j} - \alpha_{1,2k+1} + \beta_{1,2k+1} = \beta_{1,j}$ according to statement vi) of Lemma 4 with $p = 1$, $\bar{q} = i$, $j' = 2k+1$ and $j'' = j$, and $\beta_{i,2n} - \beta_{1,2n} + \beta_{1,j} = \beta_{i,j}$ due to statement viii) of Lemma 4 with $p = 1$, $\bar{q} = i$, $j' = j$ and $j'' = 2n$. □

Corollary 5 *Linear combination (29) yields equation (28).*

Proof According to Proposition 7–10, the left-hand sides of (28) and (29) are the same. Since both of them are satisfied for the points from P_n^{EVEN}, the right-hand sides also coincide with each other. □

□

5 Problem $1\,|chains, chain\text{-}length \in \{1,2\}, p_j = 1|\,\gamma$

In this section we generalize the parity inequalities (18)–(19) to the case when the precedence constraints consist of chains with length one or two.

5.1 Problem formulation

In order to simplify notation, in this section let $\mathcal{J} = \{J_1, \ldots, J_{2n+m}\}$ be the set of unit-time jobs. Let $\mathcal{C}_2 = \{C_1, \ldots, C_n\}$ be the set of chains with length 2 such that $C_i = (J_{2i-1}, J_{2i})$ for each $i \in \{1, \ldots, n\}$, that is, $J_{2i-1} \prec J_{2i}$. We say that job J_{2i-1} (J_{2i}) is the first (second) job of chain C_i. In addition, let $\mathcal{C}_1 = \{C_{n+1}, \ldots, C_{n+m}\}$ be the set of chains with length 1 such that $C_{n+i} = \{J_{2n+i}\}$ for each $i \in \{1, \ldots, m\}$. For the sake of convenience we also say that job J_{2n+i} is the first job of chain C_{n+i}.

Let $s_{i,j}$ indicate whether the first job of chain C_i is assigned to position j, for each $i \in \{1, \ldots, n+m\}$, $j \in \{1, \ldots, 2n+m\}$, and let $e_{i,j}$ indicate whether the second job of chain C_i is assigned to position j, for each $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, 2n+m\}$. We can formulate the problem as:

$$\sum_{j=1}^{2n+m} s_{i,j} = 1 \quad i \in \{1, \ldots, n+m\}, \tag{30}$$

$$\sum_{j=1}^{2n+m} e_{i,j} = 1 \quad i \in \{1, \ldots, n\}, \tag{31}$$

$$\sum_{i=1}^{n+m} s_{i,j} + \sum_{i=1}^{n} e_{i,j} = 1 \quad j \in \{1, \ldots, 2n+m\}, \tag{32}$$

$$\sum_{j=1}^{k+1} e_{i,j} \le \sum_{j=1}^{k} s_{i,j} \quad i \in \{1, \ldots, n\},\ k \in \{1, \ldots, 2n+m-1\}, \tag{33}$$

and one can strengthen the formulation by fixing some of the variables:

$$s_{i,2n+m} = 0 \quad i \in \{1, \ldots, n\}, \tag{34}$$

$$e_{i,1} = 0 \quad i \in \{1, \ldots, n\}. \tag{35}$$

5.2 Generalized parity inequalities

Given an instance $(\mathcal{J}, \mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2)$ of the problem $\Pi = (1\,|chains, chain\text{-}length \in \{1,2\}, p_j = 1|\,\gamma)$, we create an instance $(\mathcal{J}', \mathcal{C}' = \mathcal{C}_1' \cup \mathcal{C}_2')$ of the problem $\Pi' = (1\,|2\text{-}chains, p_j = 1|\,\gamma)$. On the one hand, we create m dummy jobs $J'_{2n+m+1}, \ldots, J'_{2n+2m}$ such that $\mathcal{J}' = \mathcal{J} \cup \{J'_{2n+m+1}, \ldots, J'_{2n+2m}\}$. On the other hand, let $\mathcal{C}_2' = \mathcal{C}_2$ and $\mathcal{C}_1' = \{C'_{n+1}, \ldots, C'_{n+m}\}$ where $C'_{n+i} = (J_{2n+i}, J'_{2n+m+i})$, that is, we create precedence constraints $J_{2n+i} \prec J'_{2n+m+i}$ for each $i \in \{1, \ldots, m\}$.

Consider a feasible solution for problem Π. Clearly, if we assign the dummy jobs $J'_{2n+m+1}, \ldots, J'_{2n+2m}$ arbitrary to positions $2n+m+1, \ldots, 2n+2m$ we get a feasible solution for problem Π'. Moreover, if in a feasible solution for problem Π' the jobs $J'_{2n+m+1}, \ldots, J'_{2n+2m}$ are assigned to positions $2n+m+1, \ldots, 2n+2m$, then the assignment of jobs $J_1, \ldots, J_{2n+m}$ to positions $1, \ldots, 2n+m$ yields a feasible solution for problem Π. Thus, if we consider the problem (7)–(14) for the instance $(\mathcal{J}', \mathcal{C}')$ such that $e_{n+i,2n+m+i} = 1$ for each $i \in \{1, \ldots, m\}$, then each feasible solution for that problem yields a feasible solution for the problem (30)–(33), and vice versa. Since for each $k \in \{1, \ldots, n+m-1\}$ and for each $S_2 \subseteq \{1, \ldots, n\}$, $S_1 \subseteq \{n+1, \ldots, n+m\}$ such that $S_1 \cup S_2$ has odd cardinality, the inequality

$$\sum_{j=1}^{2k} \sum_{i \in S_1 \cup S_2} (s_{i,j} - e_{i,j}) \leq |S_1 \cup S_2| - 1 + \sum_{j=1}^{2k} \sum_{i \in \bar{S}_1 \cup \bar{S}_2} (s_{i,j} - e_{i,j}), \tag{36}$$

is valid for (7)–(14) (see inequality (18)) where $\bar{S}_2 = \{1, \ldots, n\} \setminus S_2$ and $\bar{S}_1 = \{n+1, \ldots, n+m\} \setminus S_1$, thus the following inequality is valid for (30)–(33):

$$\sum_{j=1}^{2k} \left(\sum_{i \in S_2} (s_{i,j} - e_{i,j}) + \sum_{i \in S_1} s_{i,j} \right) \leq$$
$$\leq |S_1 \cup S_2| - 1 + \sum_{j=1}^{2k} \left(\sum_{i \in \bar{S}_2} (s_{i,j} - e_{i,j}) + \sum_{i \in \bar{S}_1} s_{i,j} \right). \tag{37}$$

Similarly (see inequality (19)), for each $k \in \{1, \ldots, n+m\}$ and for each $S_2 \subseteq \{1, \ldots, n\}$, $S_1 \subseteq \{n+1, \ldots, n+m\}$ such that $S_1 \cup S_2$ has even cardinality, the following inequality is valid (30)–(33):

$$\sum_{j=1}^{2k-1} \left(\sum_{i \in S_2} (s_{i,j} - e_{i,j}) + \sum_{i \in S_1} s_{i,j} \right) \leq$$
$$\leq |S_1 \cup S_2| - 1 + \sum_{j=1}^{2k-1} \left(\sum_{i \in \bar{S}_2} (s_{i,j} - e_{i,j}) + \sum_{i \in \bar{S}_1} s_{i,j} \right). \tag{38}$$

Using the transformation of problem instances mentioned above, one can use the procedure described in Section 4.4.1 to separate inequalities (37) and (38).

6 Computational experiments

In this section we present the results of our computational experiments where the main goal was to examine the effectiveness of our parity inequalities.

All the computational experiments were performed on a workstation with 8GB RAM and Intel(R) Xeon(R) CPU E5-2630 v4 of 2.20 GHz, and under Linux operating system using a single thread only. Our solution approach is implemented in C++ programming language using CPLEX (version 12.6.3.0) as the branch-and-cut framework.

In these experiments we compared three solution approaches, corresponding to the settings summarized in Table 1. Method BnC (Default) refers to the default CPLEX settings (i.e., CPLEX performs presolves and heuristics, and generates built-in cuts), while in case of methods BnB and BnC (Parity) we turned off all the presolves, heuristics and forbid to generate built-in cuts, however, in case of

Table 1 Solver settings of the different methods

Method	CPLEX Presolve	CPLEX Heuristics	CPLEX Cuts	Parity cuts
BnB	no	no	no	no
BnC (Default)	yes	yes	yes	no
BnC (Parity)	no	no	no	yes

method BnC (Parity) we separate the parity inequalities. That is, method BnB is a pure branch-and-bound procedure, and method BnC (Parity) is a pure branch-and-cut algorithm separating only the parity inequalities. In each case we solved the instances with a time limit of 600 seconds.

We generated two families of problem instances for $1\,|2\text{-}chains, p_j = 1|\,\sum w_{j,\sigma_j}$, and one family for $1\,|chains, chain\text{-}length \in \{1,2\}, p_j = 1|\,\sum w_{j,\sigma_j}$. Each family consists of 30 instances, which can be subdivided according to the number of jobs, which was $n \in \{50, 100, 150\}$, and we generated 10 instances for each n. In Tables 2–4 we summarize our results on these families, and the detailed results are presented in the appendix (see Tables 5–13). In these tables we indicate the number of jobs (n), the settings of the solver (Method), the final lower and upper bounds (LB, UB), the final gap (Gap) calculated as $100 \times (UB - LB)/LB$, the number of investigated branch-and-bound nodes (Nodes), the number of generated parity inequalities (Cuts), and the execution time (Time) in seconds.

6.1 Results on problem $1\,|2\text{-}chains, p_j = 1|\,\sum w_{j,\sigma_j}$

For the problem $1\,|2\text{-}chains, p_j = 1|\,\sum w_{j,\sigma_j}$ we generated two families of instances, Family 1 and Family 2, that differ in the method of generating the cost functions. Both families consist of 30 instances, which can be further divided into problems with $n \in \{50, 100, 150\}$ jobs, i.e., 10 instances for each n. In order to generate challenging instances, for each first-job we assigned higher weight for the early positions than for the late ones, however, for each second-job we assigned lower weight for the early positions than for the late ones. Formally, in case of Family 1, we partitioned the set of positions into 9 sets such that $\mathcal{P}_k = \{\lceil (k-1) \cdot 2n/9 \rceil + 1, \ldots, \lceil k \cdot 2n/9 \rceil\}$ for each $k \in \{1, \ldots, 9\}$, then for job J_i and position j we chose $w_{i,j}$ uniformly at random such that

- $w_{i,j} \in \{10(10-k), \ldots, 10(11-k) - 1\}$ if J_i is a first-job, and $j \in \mathcal{P}_k$,
- $w_{i,j} \in \{10k, \ldots, 10(k+1) - 1\}$ if J_i is a second-job, and $j \in \mathcal{P}_k$.

In case of Family 2, we partitioned the set of positions into 17 subsets such that $\mathcal{P}_k = \{\lceil (k-1) \cdot 2n/17 \rceil + 1, \ldots, \lceil k \cdot 2n/17 \rceil\}$ for each $k \in \{1, \ldots, 17\}$, then for job J_i and position j we chose $w_{i,j}$ uniformly at random such that

- $w_{i,j} \in \{10k, \ldots, 10(k+1) - 1\}$ if J_i is a first-job, $k \le 9$, and $j \in \mathcal{P}_k$,
- $w_{i,j} \in \{10(18-k), \ldots, 10(19-k) - 1\}$ if J_i is a first-job, $9 < k$, and $j \in \mathcal{P}_k$,
- $w_{i,j} \in \{10(10-k), \ldots, 10(11-k) - 1\}$ if J_i is a second-job, $k \le 9$, and $j \in \mathcal{P}_k$,
- $w_{i,j} \in \{10(k-9), \ldots, 10(k-8) - 1\}$ if J_i is a second-job, $9 < k$, and $j \in \mathcal{P}_k$.

In Tables 2 and 3 we summarize our results for Family 1, and Family 2, respectively, while the detailed results can be found in Tables 5-7, and in Tables 8-10,

Table 2 Summarized computational results for Family 1 (averages over 10 instances)

n	Method	LB	UB	Gap	Nodes	Cuts	Time
50	BnB	2525.4	2525.4	0.0	3196.1	0.0	17.2
	BnC (Default)	2525.4	2525.4	0.0	839.9	0.0	7.7
	BnC (Parity)	2525.4	2525.4	0.0	4.3	19.7	1.1
100	BnB	5006.9	5021.9	0.3	14 017.2	0.0	416.9
	BnC (Default)	5007.5	5011.0	0.1	10 670.4	0.0	397.2
	BnC (Parity)	5008.5	5008.5	0.0	140.0	35.7	25.0
150	BnB	7500.0	7513.7	0.2	3740.6	0.0	346.6
	BnC (Default)	7500.0	7500.1	0.0	1257.2	0.0	227.5
	BnC (Parity)	7500.0	7500.0	0.0	12.8	35.4	42.8

Table 3 Summarized computational results for Family 2 (averages over 10 instances)

n	Method	LB	UB	Gap	Nodes	Cuts	Time
50	BnB	1816.5	1816.5	0.0	925.1	0.0	3.3
	BnC (Default)	1816.5	1816.5	0.0	14.7	0.0	0.6
	BnC (Parity)	1816.5	1816.5	0.0	3.5	2.7	0.3
100	BnB	3598.6	3663.4	1.8	63 707.8	0.0	600.0
	BnC (Default)	3616.5	3644.5	0.8	14 762.8	0.0	600.0
	BnC (Parity)	3642.1	3642.1	0.0	4.3	27.1	5.4
150	BnB	5340.9	5399.8	1.1	9099.5	0.0	600.0
	BnC (Default)	5352.8	5360.5	0.1	4944.9	0.0	574.0
	BnC (Parity)	5360.4	5360.4	0.0	5.0	6.9	15.4

respectively. One can see that method BnC (Parity) significantly outperformed the other ones in all aspects. On the one hand, only the method BnC (Parity) was able to solve all instances to optimality (one can see that the average gap is always 0.0), on the other hand in each case method BnC (Parity) needed the shortest execution time, and the least number of search-tree nodes. To sum up, using parity inequalities (method BnC (Parity)) can significantly improve a pure branch-and-bound procedure (method BnB), moreover, it also outperforms the default CPLEX branch-and-cut procedure (method BnC (Default)).

6.2 Results on problem $1\,|chains, chain\text{-}length \in \{1,2\}, p_j = 1|\,\sum w_{j,\sigma_j}$

Given an n-length path (in term of number of its nodes) as the precedence graph. To obtain instances of Family 3 we randomly removed arcs from that path such that the remaining sub-paths (i.e, chains) have length at most two. For each $n \in \{50, 100, 150\}$, we generated 10 instances with n jobs, giving a total of 30 instances. Again, to generate challenging instances, for each first-job we assigned higher weight for the early positions than for the late ones, however, for each second-job we assigned lower weight for the early positions than for the late ones (see Family 1).

In Table 4 we summarize our results, and for detailed results we refer to the appendix (see Tables 11–13). Similarly to the previous experiments, the method BnC (Parity) outperformed the other ones. Namely, in case of method BnC (Parity) all

Table 4 Summarized computational results for Family 3 (averages over 10 instances)

n	Method	LB	UB	Gap	Nodes	Cuts	Time
50	BnB	2056.1	2056.1	0.0	28 892.6	0	47.0
	BnC (Default)	2056.1	2056.1	0.0	851.1	0	2.0
	BnC (Parity)	2056.1	2056.1	0.0	4.2	7.8	0.3
100	BnB	4056.1	4087.7	0.8	60 305.6	0	600.0
	BnC (Default)	4070.4	4078.7	0.2	26 954.3	0	590.3
	BnC (Parity)	4076.7	4076.7	0.0	56.0	16.0	8.6
150	BnB	6062.8	6109.5	0.8	16 331.7	0	600.0
	BnC (Default)	6068.3	6084.3	0.3	9923.6	0	600.0
	BnC (Parity)	6081.8	6081.8	0.0	32.9	16.4	23.3

instances were solved to optimality, and the separation of the parity inequalities significantly reduced the investigated branch-and-bound nodes and the execution time.

7 Conclusions, final remarks and future work

In this paper we presented polyhedral results for a single machine scheduling problem where precedence constraints are given. Among several theoretical results we also presented a class of valid inequalities that turned out to be facet defining for 2-chain precedence constraints. Our computational experiments show that separating these inequalities can significantly improve a linear programming based branch-and-bound procedure.

Our generalized inequalities are valid in the case of chain-precedence constraints when the chain-lengths are at most two. We remark that in the case of arbitrary precedence constraints one can arbitrary relax these constraints such that the remaining precedence relation consist of chains with length at most two, thus our inequalities can be separated in the general case as well. Moreover, we developed a separation algorithm that finds the most violated inequality over all the possible relaxations of the precedence constraints, however, according to computational experiments, separating these inequalities could not improve a branch-and-bound procedure.

In the future we would like to direct our attention to the case of chain precedence constraints with arbitrary chain-lengths, and to the case of general precedence constraints as well.

8 Appendix

8.1 Proof of Lemma 1

Proof (statement i)) Let $p, q \in \{1, \dots, n\}$ be distinct elements, $1 \le j_1 < j_2 < j_3 < j_4 \le 2n$ and consider points $P_1 = (s^1, e^1), P_2 = (s^2, e^2) \in P_n^{2C}$ such that $\sigma_p(P_1) = (j_1, j_2)$, $\sigma_q(P_1) = (j_3, j_4)$ and $\sigma_q(P_2) = (j_1, j_3)$, $\sigma_q(P_2) = (j_2, j_4)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, q\}$, i.e.,

$$s^1_{p,j_1} = e^1_{p,j_2} = s^1_{q,j_3} = e^1_{q,j_4} = 1 \quad \text{and} \quad s^2_{p,j_1} = e^2_{p,j_3} = s^2_{q,j_2} = e^2_{q,j_4} = 1,$$

and $s^1_{r,j} = s^2_{r,j}$, $e^1_{r,j} = e^2_{r,j}$ for all $r \notin \{p,q\}$ and $j \in \{1,\ldots,2n\}$. Since P_1 and P_2 satisfy (16), we have

$$\alpha_{p,j_1} + \beta_{p,j_2} + \alpha_{q,j_3} + \beta_{q,j_4} + \sum_{\substack{r=1\\ r\neq p,q}}^{n} \sum_{j=1}^{2n} \left(\alpha_{r,j} s^1_{r,j} + \beta_{r,j} e^1_{r,j}\right) = \gamma,$$

and

$$\alpha_{p,j_1} + \beta_{p,j_3} + \alpha_{q,j_2} + \beta_{q,j_4} + \sum_{\substack{r=1\\ r\neq p,q}}^{n} \sum_{j=1}^{2n} \left(\alpha_{r,j} s^2_{r,j} + \beta_{r,j} e^2_{r,j}\right) = \gamma,$$

thus, by subtracting the second equation from the first one, we have $\beta_{p,j_2} + \alpha_{q,j_3} = \alpha_{q,j_2} + \beta_{p,j_3}$ $(1 < j_2 < j_3 < 2n)$, that is, statement i) holds for $p \neq q$.

Since $n \geq 3$, we can choose pairwise distinct elements $p, q, r \in \{1,\ldots,n\}$, therefore we have

$$\alpha_{p,j''} - \alpha_{p,j'} = \beta_{q,j''} - \beta_{q,j'} = \alpha_{r,j''} - \alpha_{r,j'} = \beta_{p,j''} - \beta_{p,j'},$$

that is, statement i) also holds for $p = q$. □

Proof (statement ii)) Let $p, q \in \{1,\ldots,n\}$ be distinct elements, $1 \leq j_1 < j_2 < j_3 < j_4 \leq 2n$ and consider points $P_1, P_2 \in P^{2C}_n$ such that $\sigma_p(P_1) = (j_1, j_3)$, $\sigma_q(P_1) = (j_2, j_4)$ and $\sigma_p(P_2) = (j_2, j_3)$, $\sigma_q(P_2) = (j_1, j_4)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p,q\}$. Since P_1 and P_2 satisfy (16), we have $\alpha_{p,j_2} - \alpha_{p,j_1} = \alpha_{q,j_2} - \alpha_{q,j_1}$ $(1 \leq j_1 < j_2 < 2n-1)$, that is, statement ii) holds for $j'' < 2n-1$.

Now, consider points $P_3, P_4 \in P^{2C}_n$ such that $\sigma_p(P_3) = (j_1, 2n-2)$, $\sigma_q(P_3) = (2n-1, 2n)$ and $\sigma_p(P_4) = (2n-1, 2n)$, $\sigma_q(P_4) = (j_1, 2n-2)$ and $\sigma_r(P_3) = \sigma_r(P_4)$ for all $r \notin \{p,q\}$. Since P_3 and P_4 satisfy (16), we have $\alpha_{p,j_1} + \beta_{p,2n-2} + \alpha_{q,2n-1} + \beta_{q,2n} = \alpha_{p,2n-1} + \beta_{p,2n} + \alpha_{q,j_1} + \beta_{q,2n-2}$. According to statement i) (note that $1 < 2n-2$) we have $\beta_{p,2n} - \beta_{p,2n-2} = \beta_{q,2n} - \beta_{q,2n-2}$, therefore $\alpha_{p,j_1} + \alpha_{q,2n-1} = \alpha_{q,j_1} + \alpha_{p,2n-1}$, that is, statement ii) also holds for $j'' = 2n-1$. □

Proof (statement iii)) Let $p, q \in \{1,\ldots,n\}$ be distinct elements, $1 \leq j_1 < j_2 < j_3 < j_4 \leq 2n$ and consider points $P_1, P_2 \in P^{2C}_n$ such that $\sigma_p(P_1) = (j_1, j_3)$, $\sigma_q(P_1) = (j_2, j_4)$ and $\sigma_p(P_2) = (j_1, j_4)$, $\sigma_q(P_2) = (j_2, j_3)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p,q\}$. Since P_1 and P_2 satisfy (16), we have $\beta_{p,j_4} - \beta_{p,j_3} = \beta_{q,j_4} - \beta_{q,j_3}$ $(2 < j_3 < j_4 \leq 2n)$, that is, statement iii) holds for $2 < j'$.

Now, consider points $P_3, P_4 \in P^{2C}_n$ such that $\sigma_p(P_3) = (1,2)$, $\sigma_q(P_3) = (3, j_4)$ and $\sigma_p(P_4) = (3, j_4)$, $\sigma_q(P_4) = (1,2)$ and $\sigma_r(P_3) = \sigma_r(P_4)$ for all $r \notin \{p,q\}$. Since P_3 and P_4 satisfy (16), we have $\alpha_{p,1} + \beta_{p,2} + \alpha_{q,3} + \beta_{q,j_4} = \alpha_{p,3} + \beta_{p,j_4} + \alpha_{q,1} + \beta_{q,2}$. According to statement i) (note that $3 < 2n$) we have $\alpha_{p,3} - \alpha_{p,1} = \alpha_{q,3} - \alpha_{q,1}$, therefore $\beta_{p,2} + \beta_{q,j_4} = \beta_{q,2} + \beta_{p,j_4}$, that is, statement iii) also holds for $j' = 2$. □

8.2 Proof of Lemma 3

Proof (statement i)) Let $p, q \in \{1,\ldots,t\}$ be distinct elements, $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$ and consider points $P_1, P_2 \in P^{EVEN}_n$ such that $\sigma_p(P_1) = (j_1, j_3)$,

$\sigma_q(P_1) = (j_2, j_4)$ and $\sigma_p(P_2) = (j_2, j_3)$, $\sigma_q(P_2) = (j_1, j_4)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, q\}$, i.e.,

$$s^1_{p,j_1} = e^1_{p,j_3} = s^1_{q,j_2} = e^1_{q,j_4} = 1 \quad \text{and} \quad s^2_{p,j_2} = e^2_{p,j_3} = s^2_{q,j_1} = e^2_{q,j_4} = 1,$$

and $s^1_{r,j} = s^2_{r,j}$, $e^1_{r,j} = e^2_{r,j}$ for all $r \notin \{p, q\}$ and $j \in \{1, \ldots, 2n\}$. Note that such points exist according to Remark 2. Since P_1 and P_2 satisfy (28), we have

$$\alpha_{p,j_1} + \beta_{p,j_3} + \alpha_{q,j_2} + \beta_{q,j_4} + \sum_{\substack{r=1 \\ r\neq p,q}}^{n} \sum_{j=1}^{2n} \left(\alpha_{r,j} s^1_{r,j} + \beta_{r,j} e^1_{r,j} \right) = \gamma,$$

and

$$\alpha_{p,j_2} + \beta_{p,j_3} + \alpha_{q,j_1} + \beta_{q,j_4} + \sum_{\substack{r=1 \\ r\neq p,q}}^{n} \sum_{j=1}^{2n} \left(\alpha_{r,j} s^2_{r,j} + \beta_{r,j} e^2_{r,j} \right) = \gamma,$$

thus, by subtracting the first one from the second one, we have $\alpha_{p,j_1} + \alpha_{q,j_2} = \alpha_{p,j_2} + \alpha_{q,j_1}$. □

Proof (statement iii)) Let $p, q \in \{1, \ldots, t\}$ be distinct elements, $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$ and consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_1, j_3)$, $\sigma_q(P_1) = (j_2, j_4)$ and $\sigma_p(P_2) = (j_1, j_4)$, $\sigma_q(P_2) = (j_2, j_3)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, q\}$. Since P_1 and P_2 satisfy (28) we have $\beta_{p,j_4} - \beta_{p,j_3} = \beta_{q,j_4} - \beta_{q,j_3}$.

Proof (statement ii)) Let $p, q \in \{1, \ldots, t\}$ be distinct elements and $1 \leq j_1 \leq 2k < j_2 < j_3 < j_4 \leq 2n$. First, consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_1, j_3)$, $\sigma_q(P_1) = (j_2, j_4)$ and $\sigma_p(P_2) = (j_2, j_3)$, $\sigma_q(P_2) = (j_1, j_4)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, q\}$. Since P_1 and P_2 satisfy (28) we have $\alpha_{p,j_2} - \alpha_{p,j_1} = \alpha_{q,j_2} - \alpha_{q,j_1}$, that is, statement ii) holds if $2k < j'' < 2n - 1$.

Now, consider points $P_3, P_4 \in P_n^{EVEN}$ such that $\sigma_p(P_3) = (j_1, 2k+1)$, $\sigma_q(P_3) = (2n-1, 2n)$ and $\sigma_p(P_4) = (2n-1, 2n)$, $\sigma_q(P_4) = (j_1, 2k+1)$ and $\sigma_r(P_3) = \sigma_q(P_3)$ for all $r \notin \{p, q\}$. Since P_3 and P_4 satisfy (28) we have $\alpha_{p,j_1} + \beta_{p,2k+1} + \alpha_{q,2n-1} + \beta_{q,2n} = \alpha_{q,j_1} + \beta_{q,2k+1} + \alpha_{p,2n-1} + \beta_{p,2n}$. According to statement iii), $\beta_{p,2n} - \beta_{p,2k+1} = \beta_{q,2n} - \beta_{q,2k+1}$, thus $\alpha_{p,j_1} + \alpha_{q,2n-1} = \alpha_{q,j_1} + \alpha_{p,2n-1}$, that is, statement ii) also holds for $j'' = 2n - 1$. □

Proof (statement vi)) Let $p, q \in \{1, \ldots, t\}$ be distinct elements, $1 \leq j_1 < j_2 < j_3 \leq 2k < j_4 \leq 2n$. First, consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_1, j_3)$, $\sigma_q(P_1) = (j_2, j_4)$ and $\sigma_q(P_2) = (j_1, j_4)$, $\sigma_q(P_2) = (j_2, j_3)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, q\}$. Since P_1 and P_2 satisfy (28) we have $\beta_{p,j_4} - \beta_{p,j_3} = \beta_{q,j_4} - \beta_{q,j_3}$, that is, statement iv) holds if $2 < j' \leq 2k$.

Now, consider points $P_3, P_4 \in P_n^{EVEN}$ such that $\sigma_p(P_3) = (1, 2)$, $\sigma_q(P_3) = (2k, j_4)$ and $\sigma_p(P_4) = (2k, j_4)$, $\sigma_q(P_4) = (1, 2)$ and $\sigma_r(P_3) = \sigma_r(P_4)$ for all $r \notin \{p, q\}$. Since P_3 and P_4 satisfy (28) we have $\alpha_{p,1} + \beta_{p,2} + \alpha_{q,2k} + \beta_{q,j_4} = \alpha_{q,1} + \beta_{q,2} + \alpha_{p,2k} + \beta_{p,j_4}$. According to statement i), $\alpha_{p,2k} - \alpha_{p,1} = \alpha_{q,2k} - \alpha_{q,1}$, thus $\beta_{p,2} + \beta_{q,j_4} = \beta_{q,2} + \beta_{p,j_4}$, that is, statement iv) also holds for $j' = 2$. □

Proof (statement v)) Let $p, q \in \{1, \ldots, t\}$ be distinct elements, $1 \leq j_1 < j_2 < j_3 \leq 2k < j_4 \leq 2n$ and consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_1, j_3)$, $\sigma_q(P_1) = (j_2, j_4)$ and $\sigma_p(P_2) = (j_1, j_2)$, $\sigma_q(P_2) = (j_3, j_4)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, q\}$. Since P_1 and P_2 satisfy (28) we have $\alpha_{p,j_3} - \alpha_{p,j_2} = \beta_{q,j_3} - \beta_{q,j_2}$.

Since $3 \leq t$, we can choose pairwise distinct element $p, q, r \in \{1, \ldots, t\}$, therefore we have

$$\alpha_{p,j_3} - \alpha_{p,j_2} = \beta_{q,j_3} - \beta_{q,j_2} = \alpha_{r,j_3} - \alpha_{r,j_2} = \beta_{p,j_3} - \beta_{p,j_2},$$

that is, statement v) also holds for $p = q$. $\square$

Proof (statement vi)) Let $p, q \in \{1, \ldots, t\}$ be distinct elements, $1 \leq j_1 \leq 2k < j_2 < j_3 < j_4 \leq 2n$ and consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_1, j_3)$, $\sigma_q(P_1) = (j_2, j_4)$ and $\sigma_p(P_2) = (j_1, j_2)$, $\sigma_q(P_2) = (j_3, j_4)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, q\}$. Since P_1 and P_2 satisfy (28) we have $\alpha_{p,j_3} - \alpha_{p,j_2} = \beta_{q,j_3} - \beta_{q,j_2}$.

Since $3 \leq t$, we can choose pairwise distinct element $p, q, r \in \{1, \ldots, t\}$, therefore we have

$$\alpha_{p,j_3} - \alpha_{p,j_2} = \beta_{q,j_3} - \beta_{q,j_2} = \alpha_{r,j_3} - \alpha_{r,j_2} = \beta_{p,j_3} - \beta_{p,j_2},$$

that is, statement vi) also holds for $p = q$. $\square$

8.3 Proof of Lemma 4

Proof (statement vii)) Let $p \in \{1, \ldots, t\}$, $\bar{q} \in \{t+1, \ldots, n\}$ and $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_1, j_3)$, $\sigma_{\bar{q}}(P_1) = (j_2, j_4)$ and $\sigma_p(P_2) = (j_2, j_3)$, $\sigma_{\bar{q}}(P_2) = (j_1, j_4)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, \bar{q}\}$. Since P_1 and P_2 satisfy (28) we have $\alpha_{p,j_2} - \alpha_{p,j_1} = \alpha_{\bar{q},j_2} - \alpha_{\bar{q},j_1}$. $\square$

Proof (statement viii)) Let $p \in \{1, \ldots, t\}$, $\bar{q} \in \{t+1, \ldots, n\}$ and $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_1, j_3)$, $\sigma_{\bar{q}}(P_1) = (j_2, j_4)$ and $\sigma_p(P_2) = (j_1, j_4)$, $\sigma_{\bar{q}}(P_2) = (j_2, j_3)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, \bar{q}\}$. Since P_1 and P_2 satisfy (28) we have $\beta_{p,j_4} - \beta_{p,j_3} = \beta_{\bar{q},j_4} - \beta_{\bar{q},j_3}$. $\square$

Proof (statement ix)) Let $p \in \{1, \ldots, t\}$, $\bar{q} \in \{t+1, \ldots, n\}$ and $1 \leq j_1 < j_2 < j_3 \leq 2k < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_3, j_4)$, $\sigma_{\bar{q}}(P_1) = (j_1, j_2)$ and $\sigma_p(P_2) = (j_2, j_4)$, $\sigma_{\bar{q}}(P_2) = (j_1, j_3)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, \bar{q}\}$. Since P_1 and P_2 satisfy (28) we have $\alpha_{p,j_3} - \alpha_{p,j_2} = \beta_{\bar{q},j_3} - \beta_{\bar{q},j_2}$. $\square$

Proof (statement x)) Let $p \in \{1, \ldots, t\}$, $\bar{q} \in \{t+1, \ldots, n\}$ and $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_3, j_4)$, $\sigma_{\bar{q}}(P_1) = (j_1, j_2)$ and $\sigma_p(P_2) = (j_2, j_4)$, $\sigma_{\bar{q}}(P_2) = (j_1, j_3)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, \bar{q}\}$. Since P_1 and P_2 satisfy (28) we have $\alpha_{p,j_3} - \alpha_{p,j_2} = \beta_{\bar{q},j_3} - \beta_{\bar{q},j_2}$. $\square$

Proof (statement xi)) Let $p \in \{1, \ldots, t\}$, $\bar{q} \in \{t+1, \ldots, n\}$ and $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_1, j_2)$, $\sigma_{\bar{q}}(P_1) = (j_3, j_4)$ and $\sigma_p(P_2) = (j_1, j_3)$, $\sigma_{\bar{q}}(P_2) = (j_2, j_4)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, \bar{q}\}$. Since P_1 and P_2 satisfy (28) we have $\beta_{p,j_3} - \beta_{p,j_2} = \alpha_{\bar{q},j_3} - \alpha_{\bar{q},j_2}$. $\square$

Proof (statement xii)) Let $p \in \{1, \ldots, t\}$, $\bar{q} \in \{t+1, \ldots, n\}$ and $1 \leq j_1 \leq 2k < j_2 < j_3 < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_n^{EVEN}$ such that $\sigma_p(P_1) = (j_1, j_2)$, $\sigma_{\bar{q}}(P_1) = (j_3, j_4)$ and $\sigma_p(P_2) = (j_1, j_3)$, $\sigma_{\bar{q}}(P_2) = (j_2, j_4)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, \bar{q}\}$. Since P_1 and P_2 satisfy (28) we have $\beta_{p,j_3} - \beta_{p,j_2} = \alpha_{\bar{q},j_3} - \alpha_{\bar{q},j_2}$. $\square$

Table 5 Detailed computational results for Family 1 with $n = 50$

Instance	Method	LB	UB	Gap	Nodes	Cuts	Time
1	BnB	2531.0	2531.0	0.0	12 377	0	54.1
	BnC (Default)	2531.0	2531.0	0.0	535	0	13.2
	BnC (Parity)	2531.0	2531.0	0.0	4	27	1.2
2	BnB	2526.0	2526.0	0.0	520	0	6.8
	BnC (Default)	2526.0	2526.0	0.0	451	0	4.5
	BnC (Parity)	2526.0	2526.0	0.0	3	24	0.5
3	BnB	2530.0	2530.0	0.0	332	0	6.1
	BnC (Default)	2530.0	2530.0	0.0	208	0	2.8
	BnC (Parity)	2530.0	2530.0	0.0	4	17	0.8
4	BnB	2523.0	2523.0	0.0	5693	0	27.5
	BnC (Default)	2523.0	2523.0	0.0	2066	0	11.5
	BnC (Parity)	2523.0	2523.0	0.0	8	22	2.0
5	BnB	2519.0	2519.0	0.0	7	0	1.6
	BnC (Default)	2519.0	2519.0	0.0	0	0	0.8
	BnC (Parity)	2519.0	2519.0	0.0	3	17	0.7
6	BnB	2525.0	2525.0	0.0	232	0	5.6
	BnC (Default)	2525.0	2525.0	0.0	339	0	3.8
	BnC (Parity)	2525.0	2525.0	0.0	4	17	0.9
7	BnB	2527.0	2527.0	0.0	133	0	3.1
	BnC (Default)	2527.0	2527.0	0.0	14	0	0.9
	BnC (Parity)	2527.0	2527.0	0.0	5	12	1.2
8	BnB	2523.0	2523.0	0.0	1469	0	12.2
	BnC (Default)	2523.0	2523.0	0.0	624	0	4.7
	BnC (Parity)	2523.0	2523.0	0.0	5	21	1.3
9	BnB	2528.0	2528.0	0.0	8774	0	38.1
	BnC (Default)	2528.0	2528.0	0.0	3460	0	29.6
	BnC (Parity)	2528.0	2528.0	0.0	4	18	1.2
10	BnB	2522.0	2522.0	0.0	2424	0	17.0
	BnC (Default)	2522.0	2522.0	0.0	702	0	5.4
	BnC (Parity)	2522.0	2522.0	0.0	3	22	0.9
avg	BnB	2525.4	2525.4	0.0	3196.1	0.0	17.2
	BnC (Default)	2525.4	2525.4	0.0	839.9	0.0	7.7
	BnC (Parity)	2525.4	2525.4	0.0	4.3	19.7	1.1

References

R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5:287–326, 1979.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97, 1955.

J. Kulkarni and K. Munagala. Algorithms for cost-aware scheduling. In *International Workshop on Approximation and Online Algorithms*, pages 201–214. Springer, 2012.

J. K. Lenstra and A. Rinnooy Kan. Complexity results for scheduling chains on a single machine. *European Journal of Operational Research*, 4(4):270–275, 1980.

Table 6 Detailed computational results for Family 1 with $n = 100$

Instance	Method	LB	UB	Gap	Nodes	Cuts	Time
1	BnB	5006.7	5016.0	0.2	19 580	0	600.0
	BnC (Default)	5009.0	5009.0	0.0	18 317	0	513.5
	BnC (Parity)	5009.0	5009.0	0.0	160	40	25.0
2	BnB	5007.0	5007.0	0.0	3850	0	148.7
	BnC (Default)	5007.0	5007.0	0.0	2842	0	77.6
	BnC (Parity)	5007.0	5007.0	0.0	69	32	22.4
3	BnB	5008.0	5008.0	0.0	9963	0	282.5
	BnC (Default)	5008.0	5008.0	0.0	4339	0	177.8
	BnC (Parity)	5008.0	5008.0	0.0	171	36	22.9
4	BnB	5008.7	5058.0	1.0	21 653	0	600.0
	BnC (Default)	5008.5	5021.0	0.3	10 395	0	600.0
	BnC (Parity)	5013.0	5013.0	0.0	102	38	21.4
5	BnB	5006.0	5011.0	0.1	18 585	0	600.0
	BnC (Default)	5008.0	5008.0	0.0	16 913	0	524.5
	BnC (Parity)	5008.0	5008.0	0.0	263	39	37.0
6	BnB	5010.0	5010.0	0.0	5414	0	217.9
	BnC (Default)	5010.0	5010.0	0.0	3986	0	212.0
	BnC (Parity)	5010.0	5010.0	0.0	62	26	18.4
7	BnB	5005.6	5010.0	0.1	21 114	0	600.0
	BnC (Default)	5007.0	5007.0	0.0	21 630	0	483.0
	BnC (Parity)	5007.0	5007.0	0.0	221	49	29.3
8	BnB	5007.0	5007.0	0.0	1633	0	66.5
	BnC (Default)	5007.0	5007.0	0.0	11 646	0	517.4
	BnC (Parity)	5007.0	5007.0	0.0	122	19	23.5
9	BnB	5005.6	5087.0	1.6	22 484	0	600.0
	BnC (Default)	5005.9	5028.0	0.4	11 269	0	600.0
	BnC (Parity)	5011.0	5011.0	0.0	138	41	24.7
10	BnB	5005.0	5005.0	0.0	15 896	0	453.6
	BnC (Default)	5005.0	5005.0	0.0	5367	0	266.3
	BnC (Parity)	5005.0	5005.0	0.0	92	37	25.6
avg	BnB	5006.9	5021.9	0.3	14 017.2	0.0	416.9
	BnC (Default)	5007.5	5011.0	0.1	10 670.4	0.0	397.2
	BnC (Parity)	5008.5	5008.5	0.0	140.0	35.7	25.0

J. Y.-T. Leung and G. H. Young. Minimizing total tardiness on a single machine with precedence constraints. *ORSA Journal on Computing*, 2(4):346–352, 1990.

G. L. Nemhauser and L. A. Wolsey. *Integer programming and combinatorial optimization*. Wiley, Chichester, 1988.

G. Wan and X. Qi. Scheduling with variable time slot costs. *Naval Research Logistics (NRL)*, 57(2):159–171, 2010.

Y. Zhao, X. Qi, and M. Li. On scheduling with non-increasing time slot cost to minimize total weighted completion time. *Journal of Scheduling*, 19(6):759–767, 2016.

Table 7 Detailed computational results for Family 1 with $n = 150$

Instance	Method	LB	UB	Gap	Nodes	Cuts	Time
1	BnB	7500.0	7529.0	0.4	7205	0	600.0
	BnC (Default)	7500.0	7500.0	0.0	472	0	228.2
	BnC (Parity)	7500.0	7500.0	0.0	6	26	31.9
2	BnB	7500.0	7500.0	0.0	1054	0	193.7
	BnC (Default)	7500.0	7500.0	0.0	0	0	31.0
	BnC (Parity)	7500.0	7500.0	0.0	9	31	38.6
3	BnB	7500.0	7500.0	0.0	5791	0	421.3
	BnC (Default)	7500.0	7500.0	0.0	135	0	88.5
	BnC (Parity)	7500.0	7500.0	0.0	13	35	50.2
4	BnB	7500.0	7500.0	0.0	847	0	200.3
	BnC (Default)	7500.0	7500.0	0.0	2753	0	243.7
	BnC (Parity)	7500.0	7500.0	0.0	10	40	45.1
5	BnB	7500.0	7500.0	0.0	42	0	70.2
	BnC (Default)	7500.0	7500.0	0.0	2461	0	495.3
	BnC (Parity)	7500.0	7500.0	0.0	16	37	53.0
6	BnB	7500.0	7604.0	1.4	5132	0	600.0
	BnC (Default)	7500.0	7501.0	0.0	4221	0	600.0
	BnC (Parity)	7500.0	7500.0	0.0	4	78	29.2
7	BnB	7500.0	7500.0	0.0	413	0	114.7
	BnC (Default)	7500.0	7500.0	0.0	0	0	18.6
	BnC (Parity)	7500.0	7500.0	0.0	2	26	18.5
8	BnB	7500.0	7500.0	0.0	41	0	65.4
	BnC (Default)	7500.0	7500.0	0.0	246	0	89.3
	BnC (Parity)	7500.0	7500.0	0.0	9	26	38.8
9	BnB	7500.0	7502.0	0.0	7464	0	600.0
	BnC (Default)	7500.0	7500.0	0.0	785	0	129.3
	BnC (Parity)	7500.0	7500.0	0.0	9	27	41.5
10	BnB	7500.0	7502.0	0.0	9417	0	600.0
	BnC (Default)	7500.0	7500.0	0.0	1499	0	351.0
	BnC (Parity)	7500.0	7500.0	0.0	50	28	81.3
avg	BnB	7500.0	7513.7	0.2	3740.6	0.0	346.6
	BnC (Default)	7500.0	7500.1	0.0	1257.2	0.0	227.5
	BnC (Parity)	7500.0	7500.0	0.0	12.8	35.4	42.8

Table 8 Detailed computational results for Family 2 with $n = 50$

Instance	Method	LB	UB	Gap	Nodes	Cuts	Time
1	BnB	1816.0	1816.0	0.0	1978	0	5.4
	BnC (Default)	1816.0	1816.0	0.0	0	0	0.4
	BnC (Parity)	1816.0	1816.0	0.0	5	2	0.5
2	BnB	1819.0	1819.0	0.0	495	0	2.2
	BnC (Default)	1819.0	1819.0	0.0	28	0	0.8
	BnC (Parity)	1819.0	1819.0	0.0	3	3	0.3
3	BnB	1817.0	1817.0	0.0	751	0	2.3
	BnC (Default)	1817.0	1817.0	0.0	0	0	0.3
	BnC (Parity)	1817.0	1817.0	0.0	4	4	0.2
4	BnB	1814.0	1814.0	0.0	530	0	2.8
	BnC (Default)	1814.0	1814.0	0.0	0	0	0.8
	BnC (Parity)	1814.0	1814.0	0.0	4	3	0.3
5	BnB	1817.0	1817.0	0.0	845	0	3.5
	BnC (Default)	1817.0	1817.0	0.0	0	0	0.5
	BnC (Parity)	1817.0	1817.0	0.0	3	2	0.3
6	BnB	1822.0	1822.0	0.0	1835	0	5.3
	BnC (Default)	1822.0	1822.0	0.0	0	0	0.6
	BnC (Parity)	1822.0	1822.0	0.0	4	2	0.4
7	BnB	1817.0	1817.0	0.0	670	0	3.0
	BnC (Default)	1817.0	1817.0	0.0	119	0	1.2
	BnC (Parity)	1817.0	1817.0	0.0	3	2	0.2
8	BnB	1814.0	1814.0	0.0	799	0	2.8
	BnC (Default)	1814.0	1814.0	0.0	0	0	0.5
	BnC (Parity)	1814.0	1814.0	0.0	3	3	0.2
9	BnB	1819.0	1819.0	0.0	533	0	2.5
	BnC (Default)	1819.0	1819.0	0.0	0	0	0.3
	BnC (Parity)	1819.0	1819.0	0.0	3	4	0.2
10	BnB	1810.0	1810.0	0.0	815	0	2.8
	BnC (Default)	1810.0	1810.0	0.0	0	0	0.5
	BnC (Parity)	1810.0	1810.0	0.0	3	2	0.2
avg	BnB	1816.5	1816.5	0.0	925.1	0	3.3
	BnC (Default)	1816.5	1816.5	0.0	14.7	0	0.6
	BnC (Parity)	1816.5	1816.5	0.0	3.5	2.7	0.3

Table 9 Detailed computational results for Family 2 with $n = 100$

Instance	Method	LB	UB	Gap	Nodes	Cuts	Time
1	BnB	3598.5	3650.0	1.4	58 120	0	600.0
	BnC (Default)	3613.5	3644.0	0.8	15 623	0	600.0
	BnC (Parity)	3642.0	3642.0	0.0	3	24	3.4
2	BnB	3598.3	3652.0	1.5	75 054	0	600.0
	BnC (Default)	3613.2	3642.0	0.8	15 739	0	600.0
	BnC (Parity)	3641.0	3641.0	0.0	4	27	4.9
3	BnB	3599.5	3670.0	1.9	57 020	0	600.0
	BnC (Default)	3617.8	3645.0	0.8	14 084	0	600.0
	BnC (Parity)	3643.0	3643.0	0.0	4	26	5.6
4	BnB	3599.5	3655.0	1.5	73 846	0	600.0
	BnC (Default)	3617.3	3644.0	0.7	14 320	0	600.0
	BnC (Parity)	3642.0	3642.0	0.0	3	29	4.2
5	BnB	3594.5	3661.0	1.8	77 634	0	600.0
	BnC (Default)	3611.0	3643.0	0.9	14 053	0	600.0
	BnC (Parity)	3640.0	3640.0	0.0	7	31	7.1
6	BnB	3603.8	3673.0	1.9	52 608	0	600.0
	BnC (Default)	3614.0	3649.0	1.0	16 714	0	600.0
	BnC (Parity)	3646.0	3646.0	0.0	3	24	4.2
7	BnB	3596.5	3663.0	1.8	54 950	0	600.0
	BnC (Default)	3624.8	3642.0	0.5	11 172	0	600.0
	BnC (Parity)	3641.0	3641.0	0.0	3	18	4.6
8	BnB	3597.0	3667.0	1.9	53 452	0	600.0
	BnC (Default)	3628.5	3647.0	0.5	13 020	0	600.0
	BnC (Parity)	3643.0	3643.0	0.0	4	27	4.0
9	BnB	3598.5	3691.0	2.5	75 310	0	600.0
	BnC (Default)	3617.3	3646.0	0.8	14 620	0	600.0
	BnC (Parity)	3642.0	3642.0	0.0	4	19	5.4
10	BnB	3599.5	3652.0	1.4	59 084	0	600.0
	BnC (Default)	3607.5	3643.0	1.0	18 283	0	600.0
	BnC (Parity)	3641.0	3641.0	0.0	8	46	10.4
avg	BnB	3598.6	3663.4	1.8	63 707.8	0	600.0
	BnC (Default)	3616.5	3644.5	0.8	14 762.8	0	600.0
	BnC (Parity)	3642.1	3642.1	0.0	4.3	27.1	5.4

Table 10 Detailed computational results for Family 2 with $n = 150$

Instance	Method	LB	UB	Gap	Nodes	Cuts	Time
1	BnB	5342.3	5392.0	0.9	9215	0	600.0
	BnC (Default)	5357.0	5362.0	0.1	4631	0	600.0
	BnC (Parity)	5362.0	5362.0	0.0	5	6	14.1
2	BnB	5340.0	5392.0	1.0	10 210	0	600.0
	BnC (Default)	5347.0	5360.0	0.2	8701	0	600.0
	BnC (Parity)	5360.0	5360.0	0.0	4	3	14.6
3	BnB	5341.0	5386.0	0.8	9459	0	600.0
	BnC (Default)	5354.0	5360.0	0.1	5531	0	600.0
	BnC (Parity)	5360.0	5360.0	0.0	8	7	20.7
4	BnB	5341.0	5478.0	2.5	10 611	0	600.0
	BnC (Default)	5358.0	5361.0	0.1	7004	0	600.0
	BnC (Parity)	5361.0	5361.0	0.0	5	2	13.8
5	BnB	5342.0	5404.0	1.2	10 160	0	600.0
	BnC (Default)	5361.0	5361.0	0.0	2253	0	340.2
	BnC (Parity)	5361.0	5361.0	0.0	3	10	12.0
6	BnB	5340.3	5384.0	0.8	8241	0	600.0
	BnC (Default)	5347.8	5360.0	0.2	3865	0	600.0
	BnC (Parity)	5360.0	5360.0	0.0	5	9	17.2
7	BnB	5341.0	5398.0	1.1	8838	0	600.0
	BnC (Default)	5349.5	5361.0	0.2	5235	0	600.0
	BnC (Parity)	5360.0	5360.0	0.0	4	7	11.3
8	BnB	5340.0	5386.0	0.9	9313	0	600.0
	BnC (Default)	5350.5	5360.0	0.2	3570	0	600.0
	BnC (Parity)	5360.0	5360.0	0.0	7	6	18.8
9	BnB	5340.5	5377.0	0.7	7623	0	600.0
	BnC (Default)	5350.0	5360.0	0.2	3672	0	600.0
	BnC (Parity)	5360.0	5360.0	0.0	3	5	14.6
10	BnB	5341.0	5401.0	1.1	7325	0	600.0
	BnC (Default)	5353.2	5360.0	0.1	4987	0	600.0
	BnC (Parity)	5360.0	5360.0	0.0	6	14	16.5
avg	BnB	5340.9	5399.8	1.1	9099.5	0	600.0
	BnC (Default)	5352.8	5360.5	0.1	4944.9	0	574.0
	BnC (Parity)	5360.4	5360.4	0.0	5.0	6.9	15.4

Table 11 Detailed computational results for Family 3 with $n = 50$

Instance	Method	LB	UB	Gap	Nodes	Cuts	Time
1	BnB	2152.0	2152.0	0.0	41 718	0	77.2
	BnC (Default)	2152.0	2152.0	0.0	1475	0	3.2
	BnC (Parity)	2152.0	2152.0	0.0	4	6	0.4
2	BnB	2003.0	2003.0	0.0	3628	0	4.3
	BnC (Default)	2003.0	2003.0	0.0	1178	0	1.6
	BnC (Parity)	2003.0	2003.0	0.0	4	3	0.1
3	BnB	2048.0	2048.0	0.0	13 765	0	21.2
	BnC (Default)	2048.0	2048.0	0.0	534	0	2.9
	BnC (Parity)	2048.0	2048.0	0.0	3	11	0.2
4	BnB	2114.0	2114.0	0.0	15 020	0	29.8
	BnC (Default)	2114.0	2114.0	0.0	1217	0	4.0
	BnC (Parity)	2114.0	2114.0	0.0	3	9	0.3
5	BnB	2046.0	2046.0	0.0	32 895	0	52.2
	BnC (Default)	2046.0	2046.0	0.0	224	0	1.1
	BnC (Parity)	2046.0	2046.0	0.0	6	4	0.3
6	BnB	2108.0	2108.0	0.0	36 446	0	66.7
	BnC (Default)	2108.0	2108.0	0.0	282	0	1.4
	BnC (Parity)	2108.0	2108.0	0.0	8	13	0.5
7	BnB	2074.0	2074.0	0.0	99 392	0	163.1
	BnC (Default)	2074.0	2074.0	0.0	120	0	0.7
	BnC (Parity)	2074.0	2074.0	0.0	4	9	0.7
8	BnB	2001.0	2001.0	0.0	30 937	0	36.0
	BnC (Default)	2001.0	2001.0	0.0	347	0	1.0
	BnC (Parity)	2001.0	2001.0	0.0	4	6	0.2
9	BnB	1997.0	1997.0	0.0	8791	0	9.9
	BnC (Default)	1997.0	1997.0	0.0	2630	0	2.9
	BnC (Parity)	1997.0	1997.0	0.0	3	8	0.1
10	BnB	2018.0	2018.0	0.0	6334	0	9.9
	BnC (Default)	2018.0	2018.0	0.0	504	0	1.2
	BnC (Parity)	2018.0	2018.0	0.0	3	9	0.2
avg	BnB	2056.1	2056.1	0.0	28 892.6	0	47.0
	BnC (Default)	2056.1	2056.1	0.0	851.1	0	2.0
	BnC (Parity)	2056.1	2056.1	0.0	4.2	7.8	0.3

Table 12 Detailed computational results for Family 3 with $n = 100$

Instance	Method	LB	UB	Gap	Nodes	Cuts	Time
1	BnB	4158.6	4203.0	1.1	49 005	0	600.0
	BnC (Default)	4174.8	4188.0	0.3	13 032	0	600.0
	BnC (Parity)	4184.0	4184.0	0.0	17	16	8.0
2	BnB	3998.4	4018.0	0.5	57 409	0	600.0
	BnC (Default)	4001.9	4014.0	0.3	28 323	0	600.0
	BnC (Parity)	4013.0	4013.0	0.0	34	14	7.8
3	BnB	4166.1	4216.0	1.2	46 005	0	600.0
	BnC (Default)	4188.0	4195.0	0.2	12 674	0	600.0
	BnC (Parity)	4192.0	4192.0	0.0	87	7	11.3
4	BnB	4202.5	4249.0	1.1	52 099	0	600.0
	BnC (Default)	4213.6	4234.0	0.5	13 490	0	600.0
	BnC (Parity)	4229.0	4229.0	0.0	55	23	14.9
5	BnB	3997.8	4021.0	0.6	68 900	0	600.0
	BnC (Default)	4010.9	4015.0	0.1	29 273	0	600.0
	BnC (Parity)	4014.0	4014.0	0.0	43	8	7.6
6	BnB	4047.3	4077.0	0.7	57 242	0	600.0
	BnC (Default)	4072.0	4072.0	0.0	18 567	0	551.0
	BnC (Parity)	4072.0	4072.0	0.0	58	17	8.7
7	BnB	4085.3	4129.0	1.1	59 258	0	600.0
	BnC (Default)	4100.9	4116.0	0.4	12 009	0	600.0
	BnC (Parity)	4111.0	4111.0	0.0	173	24	12.5
8	BnB	3995.9	4014.0	0.5	75 567	0	600.0
	BnC (Default)	4009.8	4012.0	0.1	45 137	0	600.0
	BnC (Parity)	4012.0	4012.0	0.0	80	21	8.3
9	BnB	3955.9	3975.0	0.5	66 204	0	600.0
	BnC (Default)	3971.0	3971.0	0.0	50 761	0	551.7
	BnC (Parity)	3971.0	3971.0	0.0	6	9	3.4
10	BnB	3953.5	3975.0	0.5	71 367	0	600.0
	BnC (Default)	3961.5	3970.0	0.2	46 277	0	600.0
	BnC (Parity)	3969.0	3969.0	0.0	7	21	3.9
avg	BnB	4056.1	4087.7	0.8	60 305.6	0	600.0
	BnC (Default)	4070.4	4078.7	0.2	26 954.3	0	590.3
	BnC (Parity)	4076.7	4076.7	0.0	56.0	16	8.6

Table 13 Detailed computational results for Family 3 with $n = 150$

Instance	Method	LB	UB	Gap	Nodes	Cuts	Time
1	BnB	6332.0	6399.0	1.1	10 615	0	600.0
	BnC (Default)	6337.0	6356.0	0.3	6019	0	600.0
	BnC (Parity)	6351.0	6351.0	0.0	22	22	38.0
2	BnB	6093.0	6146.0	0.9	14 605	0	600.0
	BnC (Default)	6101.9	6115.0	0.2	10 180	0	600.0
	BnC (Parity)	6112.0	6112.0	0.0	17	9	23.6
3	BnB	6254.0	6309.0	0.9	11 203	0	600.0
	BnC (Default)	6256.0	6277.0	0.3	8483	0	600.0
	BnC (Parity)	6273.0	6273.0	0.0	7	8	16.4
4	BnB	6171.0	6244.0	1.2	13 616	0	600.0
	BnC (Default)	6173.5	6194.0	0.3	7869	0	600.0
	BnC (Parity)	6191.0	6191.0	0.0	3	21	11.2
5	BnB	5954.8	5996.0	0.7	18 052	0	600.0
	BnC (Default)	5963.3	5975.0	0.2	10 208	0	600.0
	BnC (Parity)	5973.0	5973.0	0.0	55	14	28.2
6	BnB	6017.1	6059.0	0.7	14 655	0	600.0
	BnC (Default)	6017.1	6041.0	0.4	10 356	0	600.0
	BnC (Parity)	6036.0	6036.0	0.0	60	18	27.7
7	BnB	6051.3	6087.0	0.6	17 068	0	600.0
	BnC (Default)	6057.0	6070.0	0.2	7370	0	600.0
	BnC (Parity)	6070.0	6070.0	0.0	98	22	39.4
8	BnB	5932.2	5963.0	0.5	20 137	0	600.0
	BnC (Default)	5933.0	5952.0	0.3	12 154	0	600.0
	BnC (Parity)	5951.0	5951.0	0.0	4	21	8.2
9	BnB	5891.8	5929.0	0.6	22 698	0	600.0
	BnC (Default)	5908.8	5911.0	0.0	15 587	0	600.0
	BnC (Parity)	5911.0	5911.0	0.0	6	10	8.7
10	BnB	5930.8	5963.0	0.5	20 668	0	600.0
	BnC (Default)	5935.4	5952.0	0.3	11 010	0	600.0
	BnC (Parity)	5950.0	5950.0	0.0	57	19	32.0
avg	BnB	6062.8	6109.5	0.8	16 331.7	0	600.0
	BnC (Default)	6068.3	6084.3	0.3	9923.6	0	600.0
	BnC (Parity)	6081.8	6081.8	0.0	32.9	16.4	23.3