

Annals of Operations Research (2020) 284:283–322
<https://doi.org/10.1007/s10479-019-03180-8>

ORIGINAL RESEARCH



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

Markó Horváth¹  · Tamás Kis¹ 

Published online: 1 March 2019
© The Author(s) 2019

Abstract

We consider a scheduling problem, where a set of unit-time jobs has 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 show that this problem is NP-hard even in a special case, where each chain consists of two jobs (2-chains). Further on, we study the polyhedron associated with the problem, and present a class of valid inequalities along with a polynomial-time separation procedure, and show that some of these inequalities are facet-defining in the special case of 2-chains. Finally, we present our computational results that confirm that separating these inequalities can significantly speed up a linear programming based branch-and-bound procedure to solve the problem with chains of two jobs.

Keywords Scheduling · Polyhedra · Cutting planes

1 Introduction

We consider a scheduling problem where a set of unit-time jobs has 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.

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.

✉ Tamás Kis
tamas.kis@sztaki.mta.hu

¹ Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest Kende str. 13–17, 1111, Hungary

Formally, let $\mathcal{J} = \{J_1, \dots, J_n\}$ 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 $\sigma_j^S \in \{1, \dots, n\}$ indicate 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 there is a weight $w_{j,k} \in \mathbb{Q}$, and 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 from J_i to J_j , then job J_i must be assigned to an earlier position than 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 (that is, each job has at most one immediate predecessor and at most one immediate successor), 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 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 our model, the positions of the jobs in the solution determine the job-weights in the objective function. Another, more thoroughly studied model is scheduling with position-dependent processing times, i.e., the processing time of each job is a function of its position in the sequence, see e.g., (Bachman and Janiak 2004; Rudek 2012).

In this paper we study the scheduling problem $1|chains, p_j = 1|\sum w_{j,\sigma_j}$, and provide new complexity, polyhedral, and computational results. We show that this scheduling problem is NP-hard in the strong sense, even if each chain consists of two jobs only. We also provide a natural integer programming formulation whose integer feasible solutions represent all the feasible schedules. For the corresponding polyhedron, we derive new valid inequalities strongly related to the chain structure of the precedence constraints. Our inequalities are obtained by establishing a connection to the parity polytope, investigated in Lancia and Serafini (2018). We also provide a polynomial time separation procedure. Further on, for 2-chains, i.e., where all chains consist of two jobs, we show that a sub-class of the new inequalities induces facets of the convex hull of feasible solutions of the scheduling problem. Since the problem is NP-hard in the strong sense, identifying non-trivial facets becomes even more significant. We have tested the effectiveness of our inequalities in a branch-and-cut based exact method which was implemented in C++ and tested on a number of problem instances. Our computational results show that for 2-chains, the new cuts are very effective as they accelerate the solution procedure by orders of magnitude.

The paper is organized as follows. In Sect. 3, we give an integer programming formulation for problem $1|prec, p_j = 1|\sum w_{j,\sigma_j}$, which is also appropriate for problem $1|chains, p_j = 1|\sum w_{j,\sigma_j}$. In Sect. 4, we derive valid inequalities for the convex hull of feasible solutions of the problem $1|chains, p_j = 1|\gamma$ after establishing a linear relation to the parity polytope, and we also describe a polynomial-time separation procedure. In Sect. 5 we consider a special case of $1|chains, p_j = 1|\gamma$, where each chain consists of two jobs (i.e., the precedence graph is a directed perfect matching). We denote this problem by $1|2-chains, p_j = 1|\gamma$, where the term *2-chains* indicates that each chain consists of exactly two jobs. In Sect. 5.1 we define the polytope $Q^{2-chains}$ of the feasible solutions of the class of problems $1|2-chains, p_j = 1|\gamma$. In Sect. 5.2 we prove that the problem $1|2-chains, p_j = 1|\sum w_{j,\sigma_j}$ is NP-hard in the strong sense. In Sect. 5.3, we determine the dimension of $Q^{2-chains}$, and in Sect. 5.4 we reconsider the valid inequalities derived for the general case, and we prove that some of these inequalities are facet-defining

for Q^2 -chains. Finally, in Sect. 6 we present our computational experiments, where we demonstrate that separating the facet-defining inequalities of Sect. 5.4 can significantly speed up a linear programming based branch-and-bound procedure to solve problems $1|2$ -chains, $p_j = 1|\sum w_{j,\sigma_j}$ and $1|chains, chain-length \in \{1, 2\}, p_j = 1|\sum w_{j,\sigma_j}$, where in the latter case each chain consists of one or two jobs.

2 Literature review

For a given schedule S , let C_j^S denote the completion time of a job J_j . The *makespan* of some schedule S is the maximum of the job completion times, i.e., $C_{\max}^S := \max_j C_j^S$. If a due-date d_j is given for each job J_j , then the *tardiness* of the job is $T_j^S := \max\{0, C_j^S - d_j\}$, while U_j^S indicates if the job is late, i.e., $U_j^S = 1$, if $C_j^S > d_j$, and 0 otherwise. The jobs may also have some non-negative weight w_j . The optimality criterion for minimizing the makespan, the sum of completion times, the weighted sum of completion times, the total tardiness and the throughput is denoted by C_{\max} , $\sum C_j$, $\sum w_j C_j$, $\sum T_j$ and $\sum U_j$, respectively.

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 $\gamma = \sum C_j$ are trivial (since each feasible schedule is optimal), and polynomially solvable for $\gamma = \sum w_j C_j$ [see e.g., Lawler (1978)]. Lenstra and Rinnooy Kan (1980) and Leung and Young (1990) show that problems with $\gamma = \sum U_j$ and $\gamma = \sum T_j$ are strongly 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. We summarize these results in Table 1. Although we do not consider multiple-machine scheduling problems in this paper, for the sake of completeness we also refer to some results about scheduling unit-time jobs on parallel machines under precedence constraints, i.e., problems of the form $P|prec, p_j = 1|\gamma$, where P indicates identical parallel machines. Ullman (1975) shows that problem $P|prec, p_j = 1|C_{\max}$ is strongly NP-hard, however, problems $P|chains, p_j = 1|C_{\max}$ and $P2|prec, p_j = 1|C_{\max}$ are polynomially solvable [see e.g., Hu (1961) and Coffman and Graham (1972), respectively], where $P2$ refers to the case of two parallel identical machines. Hoogeveen et al. (2001) show that problem $P|prec, p_j = 1|\sum C_j$ is APX-hard, however, problems $P|chains, p_j = 1|\sum C_j$ and $P2|prec, p_j = 1|\sum C_j$ are polynomially solvable [see e.g., Hu (1961) and Coffman and Graham (1972), respectively]. Finally, Timkovsky (2003) shows that problem $P2|chains, p_j = 1|\sum w_j C_j$ is strongly NP-hard.

The traditional precedence constraints can be considered as AND-precedence constraints, that is, a job can only be started after all of its (immediate) predecessors are completed. In contrast, in case of OR-precedence constraints, a job can be started as soon as one of its immediate predecessors is completed. Note that in this case the precedence graph can be cyclic, however, one can decide in linear time whether the problem has a feasible solution [see e.g., Möhring et al. (2004)]. According to this, problem $1|or-prec, p_j = 1|\gamma$ is trivial for $\gamma = C_{\max}$ and $\gamma = \sum C_j$, where *or-prec* refers to the presence of OR-precedence constraints. Among other results, Johannes (2005) shows that problem $1|or-prec, p_j = 1|\sum w_j C_j$ is strongly NP-hard. Note that the chain-precedence constraints are both AND- and OR-precedence constraints, since in this case each job has at most one immediate predecessor, thus problems of the form $1|chains, p_j = 1|\gamma$ considered in this paper are special cases of problem $1|or-prec, p_j = 1|\gamma$. We also summarize these results in Table 1.

Table 1 Scheduling unit-time jobs on a single machine under precedence constraints ($1|p_j, p_j = 1|\gamma$)

	$\beta = chains$	$\beta = prec$	$\beta = or-prec$
$\gamma = C_{max}$	In P (trivial) ^a		
$\gamma = \sum C_j$	In P (trivial) ^a		
$\gamma = \sum w_j C_j$	In P (Lawler 1978)	Strongly NP-hard (Lenstra and Rinnooy Kan 1978)	Strongly NP-hard (Johannes 2005)
$\gamma = \sum T_j$	Strongly NP-hard (Leung and Young 1990)		
$\gamma = \sum U_j$	Strongly NP-hard (Lenstra and Rinnooy Kan 1980)		
$\gamma = \sum w_{j,\sigma_j}$	Strongly NP-hard (in this paper)		

^aEach feasible schedule is optimal

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 k th 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 $s_j = \{t + 1, \dots, t + p_j\}$. 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, $\pi_1 \leq \dots \leq \pi_K$) 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, 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 $K = \sum_{j=1}^n p_j = n$) 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 $\mathcal{J} = \{J_1, \dots, J_n\}$ is the set of unit-time jobs, and let $\mathcal{P} = \{1, \dots, n\}$ be the set of positions. Let $D = (\mathcal{J}, A)$ be the directed acyclic precedence graph whose nodes are the jobs. We will denote by $J_{i_1} \prec\prec J_{i_2}$ if $J_{i_1} \neq J_{i_2}$ and there is a directed path from J_{i_1} to J_{i_2} in D . In this case we say that J_{i_1} is a predecessor of J_{i_2} , and J_{i_2} is a successor of J_{i_1} . Further on, we say that J_{i_1} is an immediate predecessor of J_{i_2} (denoted by $J_{i_1} \prec J_{i_2}$) if and only if $J_{i_1} \prec\prec J_{i_2}$, but there exists no job J_{i_3} such that $J_{i_1} \prec\prec J_{i_3}$ and $J_{i_3} \prec\prec J_{i_2}$. In any schedule which satisfies \prec , for each pair of jobs J_{i_1} and J_{i_1} such that $J_{i_1} \prec J_{i_2}$, job J_{i_1} must be scheduled before J_{i_2} .

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

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

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

$$\sum_{i=1}^n x_{i,j} = 1, \quad j \in \{1, \dots, n\}, \tag{3}$$

$$\sum_{j=1}^{k+1} x_{i_2,j} \leq \sum_{j=1}^k x_{i_1,j}, \quad J_{i_1} < J_{i_2}, \quad k \in \{1, \dots, n-1\}, \tag{4}$$

$$x_{i,j} \in \{0, 1\}, \quad i \in \{1, \dots, n\}, \quad j \in \{1, \dots, n\}, \tag{5}$$

where constraints (2) and (3) model the job-position assignment constraints. Constraint set (4) ensures that the precedence constraints are satisfied. That is, for each pair of jobs J_{i_1} and J_{i_2} such that $J_{i_1} < J_{i_2}$, there are $n - 1$ linear constraints ensuring that job J_{i_2} cannot be assigned to the same or to an earlier position than job J_{i_1} . Let $P_n^{prec} := \{x \in \{0, 1\}^{n \cdot n} : x \text{ satisfies (2)–(4)}\}$ be the set of the feasible solutions, and the polytope $Q_n^{prec} := \text{conv}(P_n^{prec})$ the convex hull of feasible solutions of (2)–(5). By construction, we have the following proposition.

Proposition 1 P_n^{prec} is the set of incidence vectors corresponding to feasible job-position assignments.

For later use we provide some valid equations for Q_n^{prec} . Let $\mathcal{J}_i^+ = \{J_{i'} \in \mathcal{J} : J_i << J_{i'}\}$ ($\mathcal{J}_i^- = \{J_{i'} \in \mathcal{J} : J_{i'} << J_i\}$) be the set of successors (predecessors) of job J_i . Clearly, for each point $x \in P_n^{prec}$ we have

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

$$x_{i,j} = 0, \quad i \in \{1, \dots, n\}, \quad j \in \{n - |\mathcal{J}_i^+| + 1, \dots, n\}. \tag{7}$$

Since Q_n^{prec} is the convex hull of the points P_n^{prec} , these equations are valid for Q_n^{prec} .

4 Problem 1 |chains, $p_j = 1$ | γ

In this section we present a class of valid inequalities for the case of chain-precedence constraints along with a polynomial time separation procedure. We derive these inequalities by using the so-called *parity inequalities*, which constitute the non-trivial facets of the parity polytope (see Sect. 4.1).

For chain-precedence constraints, let P_n^{chain} and Q_n^{chain} denote the set of feasible solutions and the convex hull of feasible solutions, respectively, of the integer program (1)–(5). Let $\mathcal{C} = \{C_1, \dots, C_m\}$ be the set of chains (i.e., chain-precedence constraints), where $C_i = (J_{i_1}, \dots, J_{i_\ell})$ with $J_{i_1} < \dots < J_{i_\ell}$ for each $i \in \{1, \dots, m\}$. The length of a chain C , i.e., the number of its jobs, denoted by $\text{len}(C)$. For a given integer k we denote the index set $\{1, \dots, k\}$ by $[k]$.

4.1 Parity polytope, parity inequalities

Let Q_d^{even} (Q_d^{odd}) be the convex hull of those d -dimensional 0-1 vectors in which the number of 1's is even (odd). The characterization of the parity polytope Q_d^{even} is attributed to Jeroslow (1975), however, for a direct proof of this result we refer to Lancia and Serafini (2018).

Theorem 1 (Lancia and Serafini (2018))

$$Q_d^{even} = \left\{ z \in [0, 1]^d : \sum_{i \in S} z_i - \sum_{i \notin S} z_i \leq |S| - 1, \text{ for all odd-subset } S \subseteq [d] \right\},$$

$$Q_d^{odd} = \left\{ z \in [0, 1]^d : \sum_{i \in S} z_i - \sum_{i \notin S} z_i \leq |S| - 1, \text{ for all even-subset } S \subseteq [d] \right\}.$$

We say that a subset $S \subseteq [d]$ is an odd-subset (even-subset) if its cardinality $|S|$ is odd (even), and we call the inequalities of Theorem 1 parity inequalities.

4.1.1 Separation of the parity inequalities

Since we have not been able to find any paper that provides a separation procedure for the parity inequalities, we provide our own procedure. First, we reformulate the parity inequalities as

$$1 \leq \sum_{i \in S} (1 - z_i) + \sum_{i \notin S} z_i, \text{ for each odd-subset } S \subseteq [d], \tag{8}$$

and

$$1 \leq \sum_{i \in S} (1 - z_i) + \sum_{i \notin S} z_i, \text{ for each even-subset } S \subseteq [d]. \tag{9}$$

Note that in the sake of convenience we allow S to be the complete set $[d]$, with this the corresponding inequality is still valid but redundant.

Theorem 2 *Inequalities (8) and (9) can be separated in polynomial time, that is, for a given vector $\bar{z} \in [0, 1]^d$ the following problems can be solved in polynomial time:*

$$\text{maximize } \left\{ 1 - \left(\sum_{i \in S} (1 - \bar{z}_i) + \sum_{i \notin S} \bar{z}_i \right) : S \subseteq [d] \text{ is an odd-subset} \right\}, \tag{10}$$

$$\text{maximize } \left\{ 1 - \left(\sum_{i \in S} (1 - \bar{z}_i) + \sum_{i \notin S} \bar{z}_i \right) : S \subseteq [d] \text{ is an even-subset} \right\}. \tag{11}$$

Clearly, if the maximum value is less than or equal to zero then all of the inequalities are satisfied, otherwise, the corresponding subset gives one of the most violated inequalities.

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

$$\text{minimize } \{f(S) : S \subseteq [d] \text{ is an odd-subset}\}, \tag{12}$$

$$\text{minimize } \{f(S) : S \subseteq [d] \text{ is an even-subset}\}. \tag{13}$$

- (a) *Let $S_0 := \emptyset$ and $S_i := [i]$ for all $i = 1, \dots, d$. There is an optimal solution S_{OPT} for problem (12) [problem (13)] such that $S_{OPT} = S_i$ for some $i \in \{0, \dots, d\}$.*

(b) Let $t := 0$ if $1 - v_i > v_i$ holds for all $i = 1, \dots, d$, and let $t := \max\{i : 1 - v_i \leq v_i\}$ otherwise. One of the sets S_{t-1} , S_t and S_{t+1} is an optimal solution for problem (12) [problem (13)].

Proof To prove statement (a), consider an optimal solution S_{OPT} for problem (12) which maximizes the parameter $p := \max\{i : S_i \subseteq S_{OPT}\}$, i.e., for any optimal solution S^* we have $\max\{i : S_i \subseteq S^*\} \leq p$. Clearly, $p + 1 \notin S_{OPT}$. Suppose for the sake of 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 (12), however $p < \max\{i : S_i \subseteq S'\}$ which contradicts our assumption for S_{OPT} .

According to statement (a) problems (12) and (13) can be restricted to subsets of the form S_i , $i \in \{0, \dots, d\}$. 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

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

thus if S_t has odd (even) cardinality, then it is an optimal solution for problem (12) (problem (13)), otherwise, $\arg \min\{f(S_{t-1}), f(S_{t+1})\}$ is an optimal solution for problem (12) [problem (13)]. \square

Proof (Theorem 2) For a given vector $\bar{z} \in [0, 1]^d$ let $v_i := \bar{z}_i$ for all $i = 1, \dots, d$, and let $f(S) := \sum_{i \in S} (1 - v_i) + \sum_{i \notin S} v_i$ for all $S \subseteq [d]$. Without loss of generality (e.g., by sorting and reindexing the values), we can assume that $v_1 \geq v_2 \geq \dots \geq v_d$. By this, separation problem (10) [problem (11)] is equivalent to problem (12) [problem (13)] which can be solved in polynomial time according to Lemma 1. \square

4.2 Valid inequalities for Q_n^{chain}

We introduce the variables $z_{i,j}$ ($i \in \{1, \dots, m\}$, $j \in \{1, \dots, n\}$) indicating whether the number of jobs from chain C_i that are assigned to one of the positions from $\{1, \dots, j\}$ is odd ($z_{i,j} = 1$) or even ($z_{i,j} = 0$).

Claim Let $x \in P_n^{chain}$. For each chain $C_i = (J_{i_1}, \dots, J_{i_\ell})$ and each position $j \in \{1, \dots, n\}$ we have

$$z_{i,j} = \sum_{k=1}^{\ell} (-1)^{k-1} \sum_{p=1}^j x_{i_k,p}.$$

Proof For an $x \in P_n^{chain}$ the value $\delta_k := \sum_{p=1}^j x_{i_k,p}$ ($k = 1, \dots, \ell$) equals to 1 if and only if job J_{i_k} is assigned to one of the positions $\{1, \dots, j\}$, otherwise it is 0. Clearly, for jobs $J_{i_1} < \dots < J_{i_\ell}$ we have $1 \geq \delta_1 \geq \dots \geq \delta_\ell \geq 0$, thus summing these values with alternating factors $(-1)^{k-1}$ ($k = 1, \dots, \ell$), the sum (i.e., $z_{i,j}$) is 1 if the number of δ -values that are equal to 1 is odd, otherwise it is 0. \square

Claim For an even (odd) position $j \in \{1, \dots, n\}$ the number of 1's in vector $(z_{1,j}, \dots, z_{m,j})$ is even (odd).

Proof If j is even (odd), then the number of chains C_i such that the cardinality of the set $\{k \in \{1, \dots, j\} : z_{i,k} = 1\}$ is odd (i.e., C_i has an odd number of jobs assigned to the positions $1, \dots, j$) must be even (odd). \square

According to the second claim, the corresponding parity inequalities are valid for the convex hull of the feasible solutions of the formulation extended by the z -variables. However, due to the first claim, one can transform these inequalities to the original x -variables, thus we have the following theorem.

Theorem 3 *The following inequalities are valid for Q_n^{chain} :*

$$\sum_{i \in S} \left(\sum_{k=1}^{len(C_i)} (-1)^{k-1} \sum_{p=1}^j x_{i_k,p} \right) - \sum_{i \notin S} \left(\sum_{k=1}^{len(C_i)} (-1)^{k-1} \sum_{p=1}^j x_{i_k,p} \right) \leq |S| - 1,$$

for each even position j and odd-subset $S \subseteq [m]$, (14)

and

$$\sum_{i \in S} \left(\sum_{k=1}^{len(C_i)} (-1)^{k-1} \sum_{p=1}^j x_{i_k,p} \right) - \sum_{i \notin S} \left(\sum_{k=1}^{len(C_i)} (-1)^{k-1} \sum_{p=1}^j x_{i_k,p} \right) \leq |S| - 1,$$

for each odd position j and even-subset $S \subseteq [m]$. (15)

The separation procedure of inequalities (14) [inequalities (15)] is similar to the separation procedure of inequalities (8) [inequalities (9)], that is, for a given vector $\bar{x} \in [0, 1]^{n \cdot n}$, fix an even (odd) position j , and let $\bar{z}_i := \sum_{k=1}^{\ell} (-1)^{k-1} \sum_{p=1}^j \bar{x}_{i_k,p}$ for each chain $C_i = (J_{i_1}, \dots, J_{i_\ell}), i = 1, \dots, m$. By this, one can use the separation procedure of inequalities (8) [inequalities (9)] described above.

5 Problem 1 |2-chains, $p_j = 1$ | γ

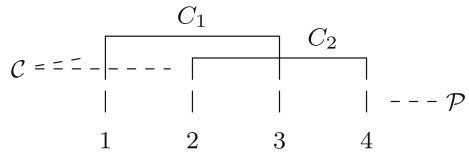
In this section we investigate the problem 1 |2-chains, $p_j = 1$ | γ . Recall that in this problem we have an even number of jobs, i.e., $2n$, and the relation $<$ partitions the set of jobs into n disjoint pairs, i.e., each jobs has exactly one predecessor or one successor, but not both. In Sect. 5.1 we reformulate the integer program of Sect. 3 to make our notation easier and reflect that each chain consists of two jobs. The problem 1 |2-chains, $p_j = 1$ | $\sum w_{j,\sigma_j}$ is shown to be strongly NP-hard in Sect. 5.2. In Sect. 5.3 we analyze the polyhedron spanned by the feasible solutions of our integer programming formulation, namely, we determine its dimension, and then in Sect. 5.4 we show that some of the inequalities from Sect. 4 are facet-defining. For basic concepts of polyhedral combinatorics we refer the reader to Nemhauser and Wolsey (1988) or Conforti et al. (2014).

5.1 Problem formulation

In order to simplify our notation, in this section let $\mathcal{J} = \{J_1, \dots, J_{2n}\}$ be the set of unit-time jobs, and $\mathcal{C} = \{C_1, \dots, C_n\}$ be the set of 2-chains, where $C_i = (J_{2i-1}, J_{2i})$, that is, $J_{2i-1} < J_{2i}$ for each $i \in \{1, \dots, n\}$. We say that job $J_{2i-1} (J_{2i})$ is the first (second) job of chain C_i . In addition, let $\mathcal{P} = \{1, \dots, 2n\}$ 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 $j \in \mathcal{P}$. Note that we just renamed the variables of the formulation (2)–(7), that is, $s_{i,j} := x_{2i-1,j}$ and $e_{i,j} := x_{2i,j}$, thus we get the following equivalent formulation:

Fig. 1 Representation of point $P = (s, e) \in P_4^{2-chains}$ with $s_{1,1} = e_{1,3} = s_{2,2} = e_{2,4} = 1$



$$\sum_{j=1}^{2n} s_{i,j} = 1, \quad i \in \{1, \dots, n\}, \tag{16}$$

$$\sum_{j=1}^{2n} e_{i,j} = 1, \quad i \in \{1, \dots, n\}, \tag{17}$$

$$s_{i,2n} = 0, \quad i \in \{1, \dots, n\}, \tag{18}$$

$$e_{i,1} = 0, \quad i \in \{1, \dots, n\}, \tag{19}$$

$$\sum_{i=1}^n s_{i,1} = 1, \tag{20}$$

$$\sum_{i=1}^n (s_{i,j} + e_{i,j}) = 1, \quad j \in \{2, \dots, 2n - 1\}, \tag{21}$$

$$\sum_{i=1}^n e_{i,2n} = 1, \tag{22}$$

$$\sum_{j=1}^{k+1} e_{i,j} \leq \sum_{j=1}^k s_{i,j}, \quad i \in \{1, \dots, n\}, \quad k \in \{1, \dots, 2n - 2\}, \tag{23}$$

$$s_{i,j}, e_{i,j} \in \{0, 1\} \quad i \in \{1, \dots, n\}, \quad j \in \{1, \dots, 2n\}. \tag{24}$$

Constraints (16)–(17) and (20)–(22) are the job-position assignment constraints [see (2) and (3)]. Constraint (23) ensures that each first-job precedes the corresponding second-job [see (4)]. Finally, constraints (18)–(19) forbid to assign a first-job to the last, or a second-job to the first position [see (6)–(7)]. Similarly to the general case in Sect. 3, we introduce the set of feasible solutions $P_{2n}^{2-chains} := \{(s, e) \in \{0, 1\}^{n \cdot (2n)} \times \{0, 1\}^{n \cdot (2n)} : (16)–(23) \text{ holds}\}$, and the polytope $Q_{2n}^{2-chains} := \text{conv}(P_{2n}^{2-chains})$.

For a given point $P = (s, e) \in P_{2n}^{2-chains}$, let $s(P, i) = j$ ($e(P, i) = j$) if $s_{i,j} = 1$ ($e_{i,j} = 1$). For a given $i \in \{1, \dots, n\}$ 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 $\sigma(P) = (\sigma_1(P), \dots, \sigma_n(P))$. For example, for the point P indicated in Fig. 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 $\sigma(P) = (1, 3, 2, 4)$.

5.2 Complexity of problem 1 |2-chains, $p_j = 1$ | $\sum w_j, \sigma_j$

In Theorem 4 we will show that problem 1 |2-chains, $p_j = 1$ | $\sum w_j, \sigma_j$ is NP-hard in the strong sense.

Sketch of proof of Theorem 4 We will transform the INDEPENDENT SET (IS) problem to problem 1 |2-chains, $p_j = 1$ | $\sum w_j, \sigma_j$. An instance of IS is given by an undirected graph

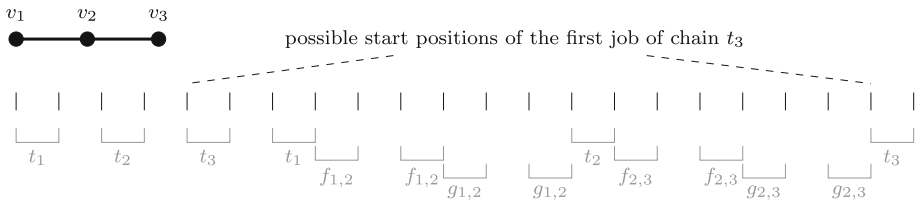


Fig. 2 Construction for the 2-length path

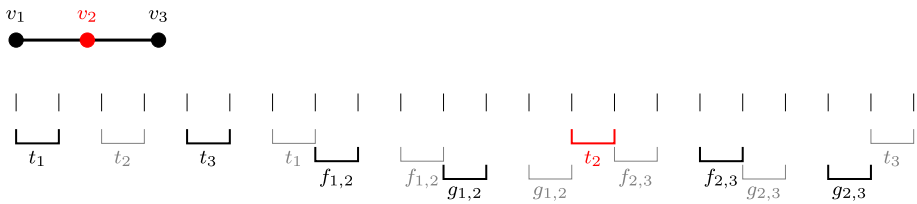


Fig. 3 Solution representing independent set $\{v_2\}$

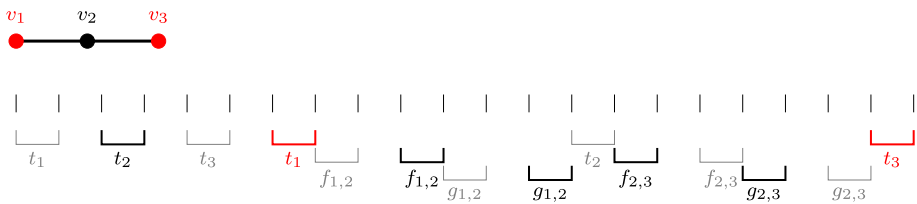


Fig. 4 Solution representing independent set $\{v_1, v_3\}$

$G = (V, E)$ with node set $V = \{v_1, \dots, v_n\}$, 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 Fig. 2, where we depict the construction for the 2-length path (without the dummy chains). 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. First, by determining appropriate weights we ensure that in each solution with non-positive total weight, each of these chains either starts and ends at its first start and end position, respectively, or at its second start and end position. In Fig. 2 we depict the two potential states of these chains. Second, by designating these positions properly, it is guaranteed that each solution with a non-positive 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 and ends its second start and end position, respectively. Note that the role of the edge-chains is to ensure that for adjacent vertices one of the corresponding node-chains must start and end at its first start and end position, respectively, i.e., at most one of these nodes can be in the independent set. For example, in Fig. 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 must 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 Fig. 4 we depict the solution that represents the independent set $\{v_1, v_3\}$ (without the dummy chains).

Theorem 4 *Problem 1 |2-chains, $p_j = 1$ | $\sum w_{j,\sigma_j}$ is NP-hard in the strong sense.*

Proof We transform the INDEPENDENT SET (IS) problem to problem 1 |2-chains, $p_j = 1$ | $\sum w_{j,\sigma_j}$. Let $G = (V, E)$ be an instance for the independent set problem with node set $V = \{v_1, \dots, v_n\}$, and edge set E , and let $\vec{E} = \{(v_i, v_j) : \{v_i, v_j\} \in E, i < j\}$ 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 $\text{succ}(i) = \{v_j : (v_i, v_j) \in \vec{E}\}$ ($\text{pred}(i) = \{v_j : (v_j, v_i) \in \vec{E}\}$) denote its immediate successors (predecessors).

Based on the IS instance we will construct an instance for problem 1 |2-chains, $p_j = 1$ | $\sum w_{j,\sigma_j}$ with $2|V| + 3|E|$ chains (that is, we will create 1 chain for each node, 2 chains for each edge, and $|V| + |E|$ additional dummy chains) and $4|V| + 6|E|$ positions.

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

To each node-chain $t_i \in \mathcal{T}_V$ we designate four distinct positions: $\alpha(t_i) < \beta(t_i) < \bar{\alpha}(t_i) < \bar{\beta}(t_i)$ such that

- (i) $2i - 1 = \alpha(t_i) = \beta(t_i) - 1$, for all $i \in \{1, \dots, n\}$,
- (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 $i \in \{1, \dots, n - 1\}$,

see Fig. 5. To each edge-chain $f_{i,j} \in \mathcal{T}_{\vec{E}}$ 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 $\text{succ}(i) = \{v_{j_1}, \dots, v_{j_{|\text{succ}(i)|}}\}$. 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 $\ell \in \{1, \dots, |\text{succ}(i)|\}$,
- (vi) $\bar{\alpha}(f_{i,j_\ell}) = \alpha(f_{i,j_{\ell+1}})$, for all $\ell \in \{1, \dots, |\text{succ}(i)| - 1\}$,
- (vii) $\bar{\alpha}(f_{i,j_{|\text{succ}(i)|}}) = \beta(f_{i,j_1}) - 1 = \bar{\beta}(f_{i,j_2}) - 2 = \dots = \bar{\beta}(f_{i,j_{|\text{succ}(i)|}}) - |\text{succ}(i)|$,
- (viii) $\bar{\beta}(f_{i,j_{|\text{succ}(i)|}}) < \bar{\alpha}(t_{i+1})$,

see Fig. 6. Finally, to each edge-chain $g_{i,j} \in \mathcal{T}_{\vec{E}}$ 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

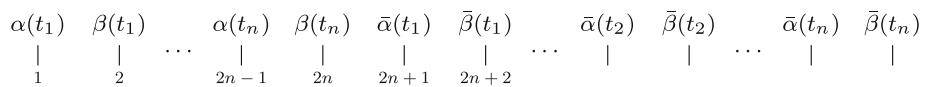


Fig. 5 Designated positions for node-chains

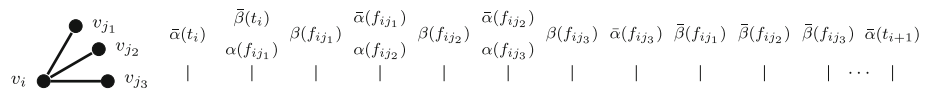


Fig. 6 Designated positions for edge-chains (part 1)

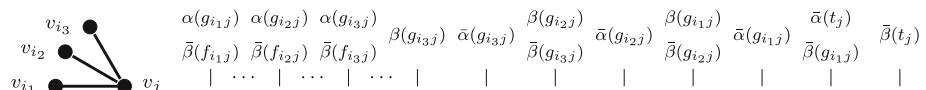


Fig. 7 Designated positions for edge-chains (part 2)

- (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 Fig. 7.

For each $v_i \in V$ we have created 1 chain and designated 4 positions, and for each $(v_i, v_j) \in \vec{E}$ we have created 2 chains and designated 8 positions, however, positions $\alpha(f_{i,j})$, $\bar{\beta}(g_{i,j})$ and $\beta(f_{i,j})$ 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 1 $|2\text{-chains}, p_j = 1| \sum w_{j,\sigma_j}$.

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

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

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

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

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

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

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

Remark that $M > n = |\mathcal{T}_V|$, thus a solution for the created problem has non-positive total weight if and only if each chain $t \in \mathcal{T}_V \cup \mathcal{T}_{\vec{E}}$ starts/ends either its first start/end or its second start/end position.

Proposition 2 Let $I \subseteq V$ an independent set in $G = (V, E)$. Then the corresponding scheduling problem instance admits a feasible solution of 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 $(v_i, v_j) \in \vec{E}$ let $s_{f_{ij},\alpha(f_{ij})} := e_{f_{ij},\beta(f_{ij})} := 1$, and for each $(v_k, v_i) \in \vec{E}$ let $s_{g_{ki},\bar{\alpha}(g_{ki})} := e_{g_{ki},\bar{\beta}(g_{ki})} := 1$ (see Fig. 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 $(v_i, v_j) \in \vec{E}$ let $s_{f_{ij},\bar{\alpha}(f_{ij})} := e_{f_{ij},\bar{\beta}(f_{ij})} := 1$, and for each $(v_k, v_i) \in \vec{E}$ let $s_{g_{ki},\alpha(g_{ki})} := e_{g_{ki},\beta(g_{ki})} := 1$ (see Fig. 9). The variables for dummy chains can be arbitrarily fixed.

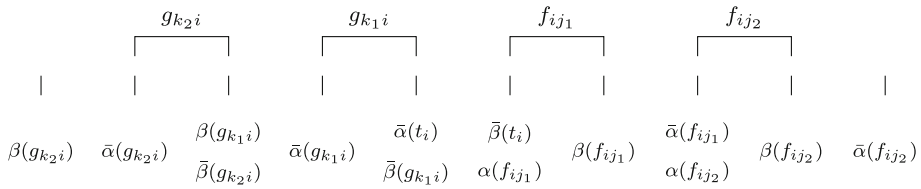


Fig. 8 Assignments for node $v_i \in V \setminus I$

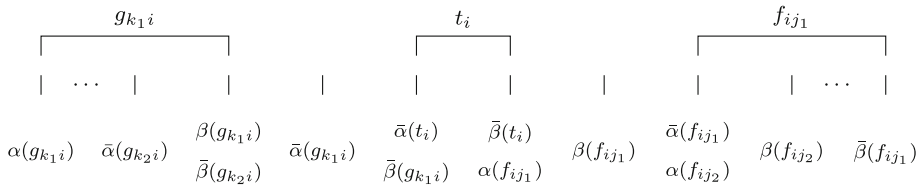


Fig. 9 Assignments for node $v_i \in I$

First, 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 $(v_i, v_j) \in \vec{E}$ [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. Second, it is clear that the weight of the solution is equal to $-|I|$. \square

Proposition 3 For an independent set problem in graph $G = (V, E)$, suppose the corresponding scheduling problem admits a feasible solution of value $W < 0$. Then there is an independent set I in G with $|I| = -W$.

Proof Since W is non-positive, according to Remark 1, for each $t \in \mathcal{T}_V \cup \mathcal{T}_{\vec{E}}$ 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 $(v_i, v_j) \in \vec{E}$ such that $v_i, v_j \in I$. Let $\text{succ}(i) = \{v_{j_1}, \dots, v_{j_{|\text{succ}(i)|}}\}$ 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 $s_{f_{ij_\ell},\bar{\alpha}(f_{ij_\ell})} = e_{f_{ij_\ell},\bar{\beta}(f_{ij_\ell})} = 1$ holds for all $\ell = 1, \dots, |\text{succ}(i)|$, moreover, since $j = j_\ell$ for some $\ell \in \{1, \dots, |\text{succ}(i)|\}$ we have $e_{f_{ij},\bar{\beta}(f_{ij})} = 1$.

Let $\text{pred}(j) = \{v_{i_1}, \dots, v_{i_{|\text{pred}(j)|}}\}$ 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 $\ell = 1, \dots, |\text{pred}(j)|$, and since $i = i_\ell$ for some $\ell \in \{1, \dots, |\text{pred}(j)|\}$ 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 $\bar{\beta}(f_{ij}) = \alpha(g_{ij})$. \square

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

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

Corollary 2 *Problem 1* $|prec, p_j = 1| \sum w_{j,\sigma_j}$ is strongly NP-hard.

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

5.3 Dimension of $Q_{2n}^{2-chains}$

In this section we investigate the dimension of the polytope $Q_{2n}^{2-chains}$.

Theorem 5

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

Sketch of the proof of Theorem 5 ($n \geq 3$) In the case of $n \geq 3$ we will apply the well-known theorem about the dimension of a non-empty polyhedron $Q = \{x \in \mathbb{R}^d : Ax \leq b\}$ claiming that $\dim(Q) + rank(E) = d$ [see e.g., Nemhauser and Wolsey (1988)], where $Ex = f$ is an equation system for Q , that is, any $x \in Q$ satisfies $Ex = f$, and if $\alpha x = \beta$ is a valid equation for Q , then there exists a vector λ of suitable dimension such that $\lambda E = \alpha$ and $\lambda f = \beta$. So, we will provide an equation system for $Q_{2n}^{2-chains}$ (see Theorem 6) with rank $6n - 1$ (see Proposition 6), which gives that the dimension of $Q_{2n}^{2-chains} \subseteq \mathbb{R}^{4n^2}$ is $4n^2 - (6n - 1)$. The detailed proof of Theorem 5 can be found at the end of Sect. 5.3.

Theorem 6 *Let* $n \geq 3$. *The equation set* $\mathcal{E} := \{(16) - (22)\}$ *is an equation system for* $Q_{2n}^{2-chains}$.

Proof Clearly, the equations of \mathcal{E} hold for every point of $Q_{2n}^{2-chains}$, since they are defining equations for this polyhedron. In order to show that \mathcal{E} is an equation system for $Q_{2n}^{2-chains}$, we show that any other equation which holds for all points of $Q_{2n}^{2-chains}$ is a linear combination of the equations of \mathcal{E} . Assume that

$$\sum_{i=1}^n \sum_{j=1}^{2n} \alpha_{i,j} s_{i,j} + \sum_{i=1}^n \sum_{j=1}^{2n} \beta_{i,j} e_{i,j} = \gamma \tag{25}$$

holds for all $(s, e) \in Q_{2n}^{2-chains}$. To show that Eq. (25) is a linear combination of Eqs. (16)–(22) we explicitly create a linear combination (26), and in Propositions 4 and 5 we prove that (25) and (26) are the same. In those proposition we use Lemma 2, however, for its proofs we refer to the ‘‘Appendix’’.

Lemma 2 *Equation (25) satisfies the following properties:*

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

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

Consider the linear combination of Eqs. (16)–(22) with coefficients $\lambda_i^{16}, \lambda_i^{17}, \lambda_i^{18}, \lambda_i^{19}, \lambda_i^{20}, \lambda_j^{21}, \lambda_j^{22}$ ($i \in \{1, \dots, n\}, j \in \{2, \dots, 2n - 1\}$), respectively, where

- $\lambda_i^{16} = \alpha_{i,1} - \alpha_{1,1}$ for all $i \in \{1, \dots, n\},$
- $\lambda_i^{17} = \beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2}$ for all $i \in \{1, \dots, n\},$

- $\lambda_i^{18} = \alpha_{i,2n} - \alpha_{i,1} + \alpha_{1,1}$ for all $i \in \{1, \dots, n\}$,
- $\lambda_i^{19} = \beta_{i,1} - \beta_{i,2n} + \beta_{1,2n} - \beta_{1,2} + \alpha_{1,2}$ for all $i \in \{1, \dots, n\}$,
- $\lambda^{20} = \alpha_{1,1}$,
- $\lambda_j^{21} = \alpha_{1,j}$ for all $j \in \{2, \dots, 2n - 1\}$,
- $\lambda^{22} = \beta_{1,2n} - \beta_{1,2} + \alpha_{1,2}$.

Let

$$\sum_{i=1}^n \sum_{j=1}^{2n} \hat{\alpha}_{i,j} s_{i,j} + \sum_{i=1}^n \sum_{j=1}^{2n} \hat{\beta}_{i,j} e_{i,j} = \hat{\gamma} \tag{26}$$

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

$$\begin{aligned} & \sum_{i=1}^n \left((\lambda_i^{16} + \lambda^{20}) s_{i,1} + (\lambda_i^{16} + \lambda_i^{18}) s_{i,2n} + (\lambda_i^{17} + \lambda_i^{19}) e_{i,1} + (\lambda_i^{17} + \lambda^{22}) e_{i,2n} \right) \\ & + \sum_{i=1}^n \sum_{j=2}^{2n-1} \left((\lambda_i^{16} + \lambda_j^{21}) s_{i,j} + (\lambda_i^{17} + \lambda_j^{21}) e_{i,j} \right). \end{aligned}$$

Proposition 4 Equation (26) satisfies the following:

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

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

$$\hat{\alpha}_{i,1} = \lambda_i^{16} + \lambda^{20} = (\alpha_{i,1} - \alpha_{1,1}) + \alpha_{1,1} = \alpha_{i,1},$$

and for $j = 2n$ we have

$$\hat{\alpha}_{i,2n} = \lambda_i^{16} + \lambda_i^{18} = (\alpha_{i,1} - \alpha_{1,1}) + (\alpha_{i,2n} - \alpha_{i,1} + \alpha_{1,1}) = \alpha_{i,2n}.$$

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

$$\hat{\alpha}_{i,j} = \lambda_i^{16} + \lambda_j^{21} = (\alpha_{i,1} - \alpha_{1,1}) + \alpha_{1,j} \stackrel{(ii)}{=} \alpha_{i,j},$$

where for the last equation we use statement (ii) of Lemma 2 with $p = 1, q = i, j' = 1$ and $j'' = j$. □

Proposition 5 For linear combination (26) the following statement holds:

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

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

$$\begin{aligned} \hat{\beta}_{i,1} &= \lambda_i^{17} + \lambda_i^{19} = (\beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2}) \\ &+ (\beta_{i,1} - \beta_{i,2n} + \beta_{1,2n} - \beta_{1,2} + \alpha_{1,2}) = \beta_{i,1}, \end{aligned}$$

and for $j = 2n$ we have

$$\hat{\beta}_{i,2n} = \lambda_i^{17} + \lambda^{22} = (\beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2}) + (\beta_{1,2n} - \beta_{1,2} + \alpha_{1,2}) = \beta_{i,2n}.$$

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

$$\hat{\beta}_{i,j} = \lambda_i^{17} + \lambda_j^{21} = (\beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2}) + \alpha_{1,j} \stackrel{(iii)}{=} \beta_{i,2} - \alpha_{1,2} + \alpha_{1,j} \stackrel{(i)}{=} \beta_{i,j}.$$

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

Corollary 4 Equation (26) is equivalent to (25).

Proof According to Propositions 4 and 5, the left-hand-sides of (26) and (25) are the same. Since both of them are satisfied by all the points in $Q_{2n}^{2-chains}$, the right-hand-sides also coincide. \square

Proposition 6 Let $n \geq 3$. The rank of the equation system $\mathcal{E} = \{(16)–(22)\}$ is $6n - 1$.

Proof Consider a linear combination of Eqs. (16)–(22) with coefficients $\lambda_i^{16}, \lambda_i^{17}, \lambda_i^{18}, \lambda_i^{19}, \lambda_i^{20}, \lambda_j^{21}, \lambda_i^{22}$ ($i \in \{1, \dots, n\}, j \in \{2, \dots, 2n - 1\}$), respectively. This linear combination can be written as

$$\begin{aligned} & \sum_{i=1}^n \left((\lambda_i^{16} + \lambda_i^{20})s_{i,1} + (\lambda_i^{16} + \lambda_i^{18})s_{i,2n} + (\lambda_i^{17} + \lambda_i^{19})e_{i,1} + (\lambda_i^{17} + \lambda_i^{22})e_{i,2n} \right) \\ & + \sum_{i=1}^n \sum_{j=2}^{2n-1} \left((\lambda_i^{16} + \lambda_j^{21})s_{i,j} + (\lambda_i^{17} + \lambda_j^{21})e_{i,j} \right) \\ & = \lambda^{20} + \lambda^{22} + \sum_{i=1}^n (\lambda_i^{16} + \lambda_i^{17}) + \sum_{j=1}^{2n} \lambda_j^{21}. \end{aligned}$$

The expression above reduces to the zero-equation ($0 \cdot s + 0 \cdot e = 0$) if and only if $\lambda_i^{18} = -\lambda_i^{16} = \lambda^{20}, \lambda_i^{16} = -\lambda_j^{21} = \lambda_i^{17}$ and $\lambda_i^{19} = -\lambda_i^{17} = \lambda^{22}$ hold for all $i \in \{1, \dots, n\}, j \in \{2, \dots, 2n - 1\}$ 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 (16)–(22), that is, we fix a single coefficient from $\lambda_i^{16}, \dots, \lambda^{22}$ to zero, then all the remaining coefficients will be zero, that is, that remaining equations are linearly independent. Hence, the equation system $\{(16)–(22)\}$ containing $6n$ equations has rank $6n - 1$. \square

Proof (Theorem 5) In case of $n = 1, P_2^{2-chains}$ consists of a single point P with $\sigma(P) = (1, 2)$, thus $\dim(Q_2^{2-chains}) = 0$.

In case of $n = 2$ in order to prove that $\dim(Q_4^{2-chains}) = 4$ we show that the maximum number of affinely independent points in $P_4^{2-chains}$ is 5. We have $P_4^{2-chains} = \{P_1, \dots, P_6\}$, 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 Fig. 10. The linear combination of these points with coefficients $\lambda_1, \dots, \lambda_6$, respectively, is

$$\begin{aligned} & (\lambda_1 + \lambda_2 + \lambda_3)s_{1,1} + (\lambda_4 + \lambda_5)s_{1,2} + \lambda_6s_{1,3} + (\lambda_4 + \lambda_5 + \lambda_6)s_{2,1} + (\lambda_2 + \lambda_3)s_{2,2} \\ & + \lambda_1s_{2,3} + \lambda_1e_{1,2} + (\lambda_2 + \lambda_4)e_{1,3} + (\lambda_3 + \lambda_5 + \lambda_6)e_{1,4} + \lambda_6e_{2,2} \\ & + (\lambda_3 + \lambda_5)e_{2,3} + (\lambda_1 + \lambda_2 + \lambda_4)e_{2,4}. \end{aligned}$$

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 such that $\lambda_1 + \dots + \lambda_6 = 0$ also holds, thus points P_1, \dots, P_6 are affinely 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 $\lambda_1 = \dots = \lambda_6 = 0$, that is, points P_1, P_3, P_4, P_5, P_6 are linearly and hence affinely independent. Therefore $\dim(Q_4^{2-chains}) = 4$.

Finally, assume that $n \geq 3$. According to Theorem 6, the equation set $\mathcal{E} = \{(16)–(22)\}$ is an equation system for $Q_{2n}^{2-chains}$, and according to Proposition 6, the rank of this system is $6n - 1$. Since we have $4n^2$ variables, thus the dimension of $Q_{2n}^{2-chains}$ is $4n^2 - (6n - 1)$. \square

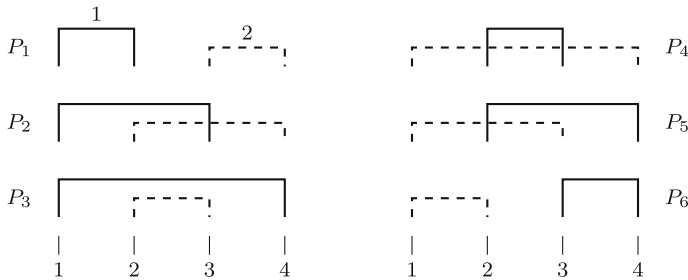


Fig. 10 The six points of $P_4^{2-chains}$

5.4 Parity inequalities

In the case of general chain-precedence constraints we showed that the parity inequalities (14) and (15) are valid for Q_n^{chain} (see Sect. 4), thus they are also valid in the case of 2-chains. Using the replacement of the variables (remark that $s_{i,j} = x_{2i-1,j}$ and $e_{i,j} = x_{2i,j}$) the following parity inequalities are valid for $Q_{2n}^{2-chains}$:

$$\sum_{i \in S} \sum_{j=1}^{2k} (s_{i,j} - e_{i,j}) - \sum_{i \notin S} \sum_{j=1}^{2k} (s_{i,j} - e_{i,j}) \leq |S| - 1,$$

for all odd-subset $S \subseteq [n]$, and $k < n$,

(27)

and

$$\sum_{i \in S} \sum_{j=1}^{2k-1} (s_{i,j} - e_{i,j}) - \sum_{i \notin S} \sum_{j=1}^{2k-1} (s_{i,j} - e_{i,j}) \leq |S| - 1,$$

for all even-subset $S \subseteq [n]$, and $k \leq n$.

(28)

In this section we show that some of the inequalities (27) are facet-defining for $Q_{2n}^{2-chains}$. Similarly, one can show that a subset of inequalities (28) are also facet-inducing.

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 $S \subseteq [n]$ with cardinality $|S| = t$. To simplify our notation, without loss of generality, we assume that $S = \{1, \dots, t\}$. The corresponding parity inequality is:

$$\sum_{i=1}^t \sum_{j=1}^{2k} (s_{i,j} - e_{i,j}) \leq t - 1 + \sum_{i=t+1}^n \sum_{j=1}^{2k} (s_{i,j} - e_{i,j}).$$
(29)

Theorem 7 *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 $S = \{1, \dots, t\}$. Inequalities (29) are facet-defining for Q_n^{chain} .*

Remark 2 Consider a point from $P_{2n}^{2-chains}$. We say that a chain $C_i = (J_{2i-1}, J_{2i})$ is active in interval $[2k, 2k + 1]$ if $\sum_{j=1}^{2k} (s_{i,j} - e_{i,j}) = 1$ holds (that is, its first job J_{2i-1} is assigned before position $2k + 1$, and its second job J_{2i} is assigned after position $2k$). A point from $P_{2n}^{2-chains}$ satisfies (29) with equality if and only if

- exactly $t - 1$ chains from $\{1, \dots, t\}$ and no chain from $\{t + 1, \dots, n\}$ are active in interval $[2k, 2k + 1]$; or

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

Sketch of the proof of Theorem 7 Let us define the set of points

$$P_{2n}^{parity} := \{(s, e) \in P_{2n}^{2-chains} : (s, e) \text{ satisfies (29) with equality}\},$$

and the polyhedron of their convex hull $Q_{2n}^{parity} := \text{conv}(P_{2n}^{parity})$. Note that Q_{2n}^{parity} is a proper face of $Q_{2n}^{2-chains}$. To prove that inequalities (29) are facet-defining for $Q_{2n}^{2-chains}$, we will show that Q_{2n}^{parity} is a facet of $Q_{2n}^{2-chains}$, i.e., $\dim(Q_{2n}^{parity}) = \dim(Q_{2n}^{2-chains}) - 1$. To do this, we apply a similar procedure as in Sect. 5.3, that is, we will prove that the set $\mathcal{E}' := \mathcal{E} \cup \{(30)\} = \{(16)–(22), (30)\}$ of equations contains a minimal equation system for Q_{2n}^{parity} with rank $\dim(Q_{2n}^{2-chains}) - 1$, where we have

$$\sum_{i=1}^t \sum_{j=1}^{2k} (s_{i,j} - e_{i,j}) + \sum_{i=t+1}^n \sum_{j=1}^{2k} (e_{i,j} - s_{i,j}) = t - 1. \tag{30}$$

The detailed proof can be found in the end of Sect. 5.4.

Theorem 8 *The equation set $\mathcal{E}' = \{(16)–(22), (30)\}$ is an equation system for Q_{2n}^{parity} .*

Proof Clearly, the equations of \mathcal{E}' hold for every point of Q_{2n}^{parity} since they are defining equations for P_{2n}^{parity} . Assume that

$$\sum_{i=1}^n \sum_{j=1}^{2n} \alpha_{i,j} s_{i,j} + \sum_{i=1}^n \sum_{j=1}^{2n} \beta_{i,j} e_{i,j} = \gamma \tag{31}$$

holds for all $(s, e) \in Q_{2n}^{parity}$. In order to show that Eq. (31) is a linear combination of Eqs. (16)–(22) and (30) we explicitly create a linear combination (32), and in Propositions 7–10 we prove that (31) and (32) are the same. In those proposition we use Lemmas 3 and 4, however for their proofs we refer to the ‘‘Appendix’’.

Lemma 3 *For Eq. (31) the following statements hold:*

- (i) $\alpha_{p,j''} - \alpha_{p,j'} = \alpha_{q,j''} - \alpha_{q,j'} \quad \forall p, q \in \{1, \dots, t\}, 1 \leq j' < j'' \leq 2k,$
- (ii) $\alpha_{p,j''} - \alpha_{p,j'} = \alpha_{q,j''} - \alpha_{q,j'} \quad \forall p, q \in \{1, \dots, t\}, 1 \leq j' \leq 2k < j'' \leq 2n - 1,$
- (iii) $\beta_{p,j''} - \beta_{p,j'} = \beta_{q,j''} - \beta_{q,j'} \quad \forall p, q \in \{1, \dots, t\}, 2k < j' < j'' \leq 2n,$
- (iv) $\beta_{p,j''} - \beta_{p,j'} = \beta_{q,j''} - \beta_{q,j'} \quad \forall p, q \in \{1, \dots, t\}, 2 \leq j' \leq 2k < j'' \leq 2n,$
- (v) $\alpha_{p,j''} - \alpha_{p,j'} = \beta_{q,j''} - \beta_{q,j'} \quad \forall p, q \in \{1, \dots, t\}, 1 < j' < j'' \leq 2k,$
- (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 Eq. (31) the following statements hold:*

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

- (x) $\alpha_{p,j''} - \alpha_{p,j'} = \beta_{\bar{q},j''} - \beta_{\bar{q},j'} \quad \forall p \in \{1, \dots, t\}, \bar{q} \in \{t+1, \dots, n\}, 1 < j' \leq 2k < j'' < 2n,$
- (xi) $\beta_{p,j''} - \beta_{p,j'} = \alpha_{\bar{q},j''} - \alpha_{\bar{q},j'} \quad \forall p \in \{1, \dots, t\}, \bar{q} \in \{t+1, \dots, n\}, 1 < j' \leq 2k < j'' < 2n,$
- (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 Eqs. (16)–(22) and (30) with coefficients $\lambda_i^{16}, \lambda_i^{17}, \lambda_i^{18}, \lambda_i^{19}, \lambda^{20}, \lambda_j^{21}, \lambda^{22}$ and $\lambda^{30}, (i \in \{1, \dots, n\}, j \in \{2, \dots, 2n-1\})$ respectively, where

- $\lambda^{30} = \lambda$, where $\lambda := (\alpha_{1,2} - \alpha_{1,2k+1} - \beta_{1,2} + \beta_{1,2k+1})/2,$
- $\lambda_i^{16} = \begin{cases} \alpha_{i,1} - \alpha_{1,1} & \text{if } i \in \{1, \dots, t\}, \\ \alpha_{i,1} - \alpha_{1,1} + 2\lambda & \text{if } i \in \{t+1, \dots, n\}, \end{cases}$
- $\lambda_i^{17} = \mu_i$, where $\mu_i := \begin{cases} \beta_{1,2} - \alpha_{1,2} + 2\lambda & \text{if } i = 1, \\ \beta_{i,2n} - \beta_{1,2n} + \mu_1 & \text{if } i \in \{2, \dots, n\}, \end{cases}$
- $\lambda_i^{18} = \begin{cases} \alpha_{1,2n} & \text{if } i = 1, \\ \alpha_{i,2n} - \alpha_{i,1} + \alpha_{1,1} & \text{if } i \in \{2, \dots, t\}, \\ \alpha_{i,2n} - \alpha_{i,1} + \alpha_{1,1} - 2\lambda & \text{if } i \in \{t+1, \dots, n\}, \end{cases}$
- $\lambda_i^{19} = \begin{cases} \beta_{i,1} + \lambda - \mu_i & \text{if } i \in \{1, \dots, t\}, \\ \beta_{i,1} - \lambda - \mu_i & \text{if } i \in \{t+1, \dots, n\}, \end{cases}$
- $\lambda^{20} = \alpha_{1,1} - \lambda,$
- $\lambda_j^{21} = \begin{cases} \alpha_{1,j} - \lambda & \text{if } j \in \{2, \dots, 2k\}, \\ \alpha_{1,j} & \text{if } j \in \{2k+1, \dots, 2n-1\}, \end{cases}$
- $\lambda^{22} = \beta_{1,2n} - \mu_1.$

Let

$$\sum_{i=1}^n \sum_{j=1}^{2n} \hat{\alpha}_{i,j} s_{i,j} + \sum_{i=1}^n \sum_{j=1}^{2n} \hat{\beta}_{i,j} e_{i,j} = \hat{\gamma} \tag{32}$$

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

$$\begin{aligned} & \sum_{i=1}^t \left((\lambda_i^{16} + \lambda^{20} + \lambda^{30}) s_{i,1} + (\lambda_i^{17} + \lambda_i^{19} - \lambda^{30}) e_{i,1} \right) \\ & + \sum_{i=t+1}^n \left((\lambda_i^{16} + \lambda^{20} - \lambda^{30}) s_{i,1} + (\lambda_i^{17} + \lambda_i^{19} + \lambda^{30}) e_{i,1} \right) \\ & + \sum_{i=1}^n \left((\lambda_i^{16} + \lambda_i^{18}) s_{i,2n} + (\lambda_i^{17} + \lambda^{22}) e_{i,2n} \right) \\ & + \sum_{i=1}^t \sum_{j=2}^{2k} \left((\lambda_i^{16} + \lambda_j^{21} + \lambda^{30}) s_{i,j} + (\lambda_i^{17} + \lambda_j^{21} - \lambda^{30}) e_{i,j} \right) \\ & + \sum_{i=t+1}^n \sum_{j=2}^{2k} \left((\lambda_i^{16} + \lambda_j^{21} - \lambda^{30}) s_{i,j} + (\lambda_i^{17} + \lambda_j^{21} + \lambda^{30}) e_{i,j} \right) \\ & + \sum_{i=1}^n \sum_{j=2k+1}^{2n-1} \left((\lambda_i^{16} + \lambda_j^{21}) s_{i,j} + (\lambda_i^{17} + \lambda_j^{21}) e_{i,j} \right). \end{aligned}$$

Proposition 7 For linear combination (32) 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

$$\hat{\alpha}_{i,1} = \lambda_i^{16} + \lambda^{20} + \lambda^{30} = (\alpha_{i,1} - \alpha_{1,1}) + (\alpha_{1,1} - \lambda) + \lambda = \alpha_{i,1},$$

and for $j = 2n$ we have

$$\hat{\alpha}_{i,2n} = \lambda_i^{16} + \lambda_i^{18} = (\alpha_{i,1} - \alpha_{1,1}) + (\alpha_{i,2n} + \alpha_{1,1} - \alpha_{i,1}) = \alpha_{i,2n}.$$

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

$$\hat{\alpha}_{i,j} = \lambda_i^{16} + \lambda_j^{21} + \lambda^{30} = (\alpha_{i,1} - \alpha_{1,1}) + (\alpha_{1,j} - \lambda) + \lambda = \alpha_{1,j} - \alpha_{1,1} + \alpha_{i,1} \stackrel{(i)}{=} \alpha_{i,j},$$

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 $j \in \{2k + 1, \dots, 2n - 1\}$ we have

$$\hat{\alpha}_{i,j} = \lambda_i^{16} + \lambda_j^{21} = \alpha_{1,j} - \alpha_{1,1} + \alpha_{i,1} \stackrel{(ii)}{=} \alpha_{i,j},$$

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 (32) the following statement holds:

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

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

$$\hat{\beta}_{1,1} = \lambda_1^{17} + \lambda_1^{19} - \lambda^{30} = \mu_1 + (\beta_{1,1} + \lambda - \mu_1) - \lambda = \beta_{1,1},$$

and for $j = 2n$ we have

$$\hat{\beta}_{1,2n} = \lambda_1^{17} + \lambda^{22} = \mu_1 + (\beta_{1,2n} - \mu_1) = \beta_{1,2n}.$$

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

$$\begin{aligned} \hat{\beta}_{1,j} &= \lambda_1^{17} + \lambda_j^{21} - \lambda^{30} = (\beta_{1,2} - \alpha_{1,2} + 2\lambda) + (\alpha_{1,j} - \lambda) - \lambda \\ &= \beta_{1,2} + \alpha_{1,j} - \alpha_{1,2} \stackrel{(v)}{=} \beta_{1,j}, \end{aligned}$$

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 $j \in \{2k + 1, \dots, 2n - 1\}$ we have

$$\hat{\beta}_{1,j} = \lambda_1^{17} + \lambda_j^{21} = \beta_{1,2} + \alpha_{1,j} - \alpha_{1,2} + 2\lambda = \alpha_{1,j} - \alpha_{1,2k+1} + \beta_{1,2k+1} \stackrel{(vi)}{=} \beta_{i,j},$$

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

$$\hat{\beta}_{i,1} = \lambda_i^{17} + \lambda_i^{19} - \lambda^{30} = \mu_i + (\beta_{i,1} + \lambda - \mu_i) - \lambda = \beta_{i,1},$$

and for $j = 2n$ we have

$$\hat{\beta}_{i,2n} = \lambda_i^{17} + \lambda^{22} = (\beta_{i,2n} - \beta_{1,2n} + \mu_1) + (\beta_{1,2n} - \mu_1) = \beta_{i,2n}.$$

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

$$\begin{aligned} \hat{\beta}_{i,j} &= \lambda_i^{17} + \lambda_j^{21} - \lambda^{30} = (\beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2} + 2\lambda) + (\alpha_{1,j} - \lambda) - \lambda \\ &= \beta_{i,2n} - \beta_{1,2n} + \beta_{1,2} - \alpha_{1,2} + \alpha_{1,j} \stackrel{(v)}{=} \beta_{i,2n} - \beta_{1,2n} + \beta_{1,j} \stackrel{(iv)}{=} \beta_{i,j}, \end{aligned}$$

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 $j \in \{2k + 1, \dots, 2n - 1\}$ we have

$$\begin{aligned} \hat{\beta}_{i,j} &= \lambda_i^{17} + \lambda_j^{21} = \beta_{i,2n} - \beta_{1,2n} + \alpha_{1,j} - \alpha_{1,2k+1} \\ &\quad + \beta_{1,2k+1} \stackrel{(vi)}{=} \beta_{i,2n} - \beta_{1,2n} + \beta_{1,j} \stackrel{(iv)}{=} \beta_{i,j}, \end{aligned}$$

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 (32) 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

$$\hat{\alpha}_{i,1} = \lambda_i^{16} + \lambda^{20} - \lambda^{30} = (\alpha_{i,1} - \alpha_{1,1} + 2\lambda) + (\alpha_{1,1} - \lambda) - \lambda = \alpha_{i,1},$$

and for $j = 2n$ we have

$$\hat{\alpha}_{i,2n} = \lambda_i^{16} + \lambda_i^{18} = (\alpha_{i,1} - \alpha_{1,1} + 2\lambda) + (\alpha_{i,2n} - \alpha_{i,1} + \alpha_{1,1} - 2\lambda) = \alpha_{i,2n}.$$

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

$$\begin{aligned} \hat{\alpha}_{i,j} &= \lambda_i^{16} + \lambda_j^{21} - \lambda^{30} = (\alpha_{i,1} - \alpha_{1,1} + 2\lambda) + (\alpha_{1,j} - \lambda) - \lambda \\ &= \alpha_{1,j} - \alpha_{1,1} + \alpha_{i,1} \stackrel{(vii)}{=} \alpha_{i,j}, \end{aligned}$$

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

$$\begin{aligned} \hat{\alpha}_{i,j} &= \lambda_i^{16} + \lambda_j^{21} = (\alpha_{i,1} - \alpha_{1,1} + \alpha_{1,2} - \beta_{1,2} + \beta_{1,2k+1} - \alpha_{1,2k+1}) + \alpha_{1,j} \\ &\stackrel{(vi)}{=} \alpha_{i,1} - \alpha_{1,1} + \alpha_{1,2} - \beta_{1,2} + \beta_{1,j} \stackrel{(vii)}{=} \alpha_{i,2} - \beta_{1,2} + \beta_{1,j} \stackrel{(xi)}{=} \alpha_{i,j}, \end{aligned}$$

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 (32) 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

$$\hat{\beta}_{i,1} = \lambda_i^{17} + \lambda_i^{19} + \lambda^{30} = \mu_i + (\beta_{i,1} - \lambda - \mu_i) + \lambda = \beta_{i,1},$$

and for $j = 2n$ we have

$$\hat{\beta}_{1,2n} = \lambda_i^{17} + \lambda^{22} = (\beta_{i,2n} - \beta_{1,2n} + \mu_1) + (\beta_{1,2n} - \mu_1) = \beta_{i,2n}.$$

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

$$\begin{aligned} \hat{\beta}_{i,j} &= \lambda_i^{17} + \lambda_j^{21} + \lambda^{30} = \beta_{i,2n} + \alpha_{1,j} - \beta_{1,2n} - \alpha_{1,2k+1} + \beta_{1,2k+1} \\ &\stackrel{(viii)}{=} \alpha_{1,j} - \alpha_{1,2k+1} + \beta_{i,2k+1} \stackrel{(x)}{=} \beta_{i,j}, \end{aligned}$$

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

$$\begin{aligned} \hat{\beta}_{i,j} &= \lambda_i^{17} + \lambda_j^{21} = \beta_{i,2n} - \beta_{1,2n} + \alpha_{1,j} \\ &\quad - \alpha_{1,2k+1} + \beta_{1,2k+1} \stackrel{(vi)}{=} \beta_{i,2n} - \beta_{1,2n} + \beta_{1,j} \stackrel{(viii)}{=} \beta_{i,j}, \end{aligned}$$

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 (32) yields Eq. (31).*

Proof According to Propositions 7–10, the left-hand sides of (31) and (32) are the same. Since both of them are satisfied for the points from P_{2n}^{parity} , the right-hand sides also coincide with each other. □

Proof (Theorem 7) First, by definition, $rank(\mathcal{E}') \leq rank(\mathcal{E}) + 1$, thus (according to Theorems 6 and 8) $\dim(Q_{2n}^{parity}) \geq \dim(Q_{2n}^{2-chains}) - 1$. Second, $\dim(Q_{2n}^{parity}) \leq \dim(Q_{2n}^{2-chains}) - 1$ since Q_{2n}^{parity} is a proper face of $Q_{2n}^{2-chains}$, thus $\dim(Q_{2n}^{parity}) = \dim(Q_{2n}^{2-chains}) - 1$ and Q_{2n}^{parity} is a facet of $Q_{2n}^{2-chains}$. □

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. Since we proved that some of these inequalities are facet-defining if each chain has length two, our experiments focused on problems $1|2-chains, p_j = 1| \sum w_{j,\sigma_j}$ and $1|chain-length \in \{1, 2\}, p_j = 1| \sum w_{j,\sigma_j}$, where in the latter case each chain has length at most two.

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, more precisely, four scenarios corresponding to the settings summarized in Table 2. Method BnB is pure branch-and-bound, where we turned off all the presolves, heuristics and forbid to generate built-in cuts. Method BnC (Default) refers to the default CPLEX settings (i.e., CPLEX performs presolves and heuristics, and generates built-in cuts). Methods BnC (Parity)-1 and BnC (Parity)-2 use the same solver settings as BnB, but we also separate parity inequalities, i.e., both of

Table 2 Solver settings of the different methods

Method	CPLEX			Parity cuts	
	Presolve	Heuristics	Cuts	Root	Non-root
BnB	No	No	No	No	No
BnC (Default)	Yes	Yes	Yes	No	No
BnC (Parity)-1	No	No	No	No	Yes
BnC (Parity)-2	No	No	No	Yes	Yes

these methods separate parity inequalities in search-tree node of depth at least 1, but BnC (Parity)-1 does not generate any cuts in the root, while BnC (Parity)-2 does. In each case we had a runtime limit of 600 s, i.e., the search was stopped upon reaching the time limit.

We generated two families of problem instances for 1 |2-chains, $p_j = 1$ | $\sum w_{j,\sigma_j}$, and one family for 1 |chains, $chain-length \in \{1, 2\}$, $p_j = 1$ | $\sum w_{j,\sigma_j}$. 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 3, 4 and 5 we summarize our results on these families, and the detailed results are presented in the “Appendix” (see Tables 6, 7, 8, 9, 10, 11, 12, 13 and 14). In these tables we indicate the number of jobs (n), the settings of the solver (Method), the lower bound after the root node is solved (LB^r), the final lower and upper bounds (LB^f, UB^f), the final gap (Gap^f) calculated as $100 \times (UB^f - LB^f)/LB^f$, 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-chains, $p_j = 1$ | $\sum w_{j,\sigma_j}$

For the problem 1 |2-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 $\mathcal{P}_k = \{[(k - 1) \cdot 2n/9] + 1, \dots, [k \cdot 2n/9]\}$ for each $k \in \{1, \dots, 9\}$, then for job J_i and position j we chose $w_{i,j}$ uniformly at random such that

- $w_{i,j} \in \{10(10 - k), \dots, 10(11 - k) - 1\}$ if J_i is a first-job, and $j \in \mathcal{P}_k$,
- $w_{i,j} \in \{10k, \dots, 10(k + 1) - 1\}$ 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 $\mathcal{P}_k = \{[(k - 1) \cdot 2n/17] + 1, \dots, [k \cdot 2n/17]\}$ for each $k \in \{1, \dots, 17\}$, then for job J_i and position j we chose $w_{i,j}$ uniformly at random such that

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

In Tables 3 and 4 we summarize our results for Family 1 and Family 2, respectively, while the detailed results can be found in Tables 6, 7 and 8, and in Tables 9, 10 and 11, respectively. Our observations are the followings.

Table 3 Summarized computational results for Family 1 (averages over 10 instances)

n	Method	LB^r	LB^f	UB^f	Gap^f	Nodes	Cuts	Time
50	BnB	2521.1	2525.4	2525.4	0.0	3196.1	0.0	17.2
	BnC (Default)	2522.3	2525.4	2525.4	0.0	839.9	0.0	7.7
	BnC (Parity)-1	2521.1	2525.4	2525.4	0.0	4.3	19.7	1.1
	BnC (Parity)-2	2525.0	2525.4	2525.4	0.0	2.3	20.2	0.6
100	BnB	5004.8	5006.9	5021.9	0.3	14, 017.2	0.0	416.9
	BnC (Default)	5005.1	5007.5	5011.0	0.1	10, 670.4	0.0	397.2
	BnC (Parity)-1	5004.8	5008.5	5008.5	0.0	140.0	35.7	25.0
	BnC (Parity)-2	5006.9	5008.5	5008.5	0.0	127.6	26.4	23.3
150	BnB	7500.0	7500.0	7513.7	0.2	3740.6	0.0	346.6
	BnC (Default)	7500.0	7500.0	7500.1	0.0	1257.2	0.0	227.5
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	12.8	35.4	42.8
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	82.8	26.7	82.0

- Methods BnC (Parity)-1 and BnC (Parity)-2 significantly outperformed the other ones in all aspects. First, only these methods were able to solve all instances to optimality (one can see that the average gap is always 0.0), Second, for each instance, method BnC (Parity)-1 needed shorter execution time than methods BnB and BnC (Default). Note that on average, method BnC (Parity)-2 was also significantly faster than methods BnB and BnC (Default) [often faster than method BnC (Parity)-1 as well], however, for some instances one of the other two methods outperformed it. Finally, both of the methods BnC (Parity)-1 and BnC (Parity)-2 significantly reduced the number of the explored tree-nodes as well.
- Separating parity inequalities at the root node [method BnC (Parity)-2] yielded the best (i.e., highest) lower bounds at the root node, however, on large instances with 150 jobs the separation procedure at the root node took a lot of time, which resulted in longer execution times than the method BnC (Parity)-1. For example, in case of Family 1 and $n = 150$, where the LP-relaxation of the problem (see column LB^r of the pure branch-and-bound method BnB) is basically strong, separating these inequalities at the root node could not help a lot, and method BnC (Parity)-1 outperformed method BnC (Parity)-2.

To sum up, using parity inequalities [methods BnC (Parity)-1 and BnC (Parity)-2] can significantly improve a pure branch-and-bound procedure (method BnB), moreover, they also outperform the default CPLEX branch-and-cut procedure [method BnC (Default)].

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

Given an n -length path (in terms 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).

Table 4 Summarized computational results for Family 2 (averages over 10 instances)

n	Method	LB^r	LB^f	UB^f	Gap^f	Nodes	Cuts	Time
50	BnB	1800.4	1816.5	1816.5	0.0	925.1	0.0	3.3
	BnC (Default)	1815.9	1816.5	1816.5	0.0	14.7	0.0	0.6
	BnC (Parity)-1	1800.4	1816.5	1816.5	0.0	3.5	2.7	0.3
	BnC (Parity)-2	1816.3	1816.5	1816.5	0.0	1.7	2.6	0.2
100	BnB	3596.2	3598.6	3663.4	1.8	63, 707.8	0.0	600.0
	BnC (Default)	3602.3	3616.5	3644.5	0.8	14, 762.8	0.0	600.0
	BnC (Parity)-1	3596.2	3642.1	3642.1	0.0	4.3	27.1	5.4
	BnC (Parity)-2	3642.1	3642.1	3642.1	0.0	4.5	16.9	5.2
150	BnB	5340.7	5340.9	5399.8	1.1	9099.5	0.0	600.0
	BnC (Default)	5344.3	5352.8	5360.5	0.1	4944.9	0.0	574.0
	BnC (Parity)-1	5340.7	5360.4	5360.4	0.0	5.0	6.9	15.4
	BnC (Parity)-2	5360.4	5360.4	5360.4	0.0	6.8	10.0	18.1

Table 5 Summarized computational results for Family 3 (averages over 10 instances)

n	Method	LB^r	LB^f	UB^f	Gap^f	Nodes	Cuts	Time
50	BnB	2039.6	2056.1	2056.1	0.0	28, 892.6	0.0	47.0
	BnC (Default)	2049.8	2056.1	2056.1	0.0	851.1	0.0	2.0
	BnC (Parity)-1	2039.6	2056.1	2056.1	0.0	4.2	7.8	0.3
	BnC (Parity)-2	2055.6	2056.1	2056.1	0.0	2.9	5.3	0.3
100	BnB	4053.6	4056.1	4087.7	0.8	60, 305.6	0.0	600.0
	BnC (Default)	4056.8	4070.4	4078.7	0.2	26, 954.3	0.0	590.3
	BnC (Parity)-1	4053.6	4076.7	4076.7	0.0	56.0	16.0	8.6
	BnC (Parity)-2	4075.7	4076.7	4076.7	0.0	44.4	8.5	6.9
150	BnB	6062.4	6062.8	6109.5	0.8	16, 331.7	0.0	600.0
	BnC (Default)	6063.7	6068.3	6084.3	0.3	9923.6	0.0	600.0
	BnC (Parity)-1	6062.4	6081.8	6081.8	0.0	32.9	16.4	23.3
	BnC (Parity)-2	6081.4	6081.8	6081.8	0.0	196.9	10.6	38.3

In Table 5 we summarize our results, and for detailed results we refer to the “Appendix” (see Tables 12, 13 and 14). Similarly to the previous experiments, the methods BnC (Parity)-1 and BnC (Parity)-2 outperformed the other ones. For smaller instances with 50 and 100 jobs, BnC (Parity)-2 slightly outperformed BnC (Parity)-1 in terms of search-tree nodes and total running time, but on large instances with 150 jobs, the BnC (Parity)-1 proved better.

7 Conclusions, final remarks and future work

In this paper we presented polyhedral and complexity 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-chains precedence constraints. Our computational experiments show that separating these inequalities

can significantly improve a linear programming based branch-and-bound procedure if the length of each chain is at most two. Although these inequalities are also valid in the case of chain-precedence constraints with arbitrary chain-lengths, according to our preliminary computational experiments, separating these inequalities could not improve a branch-and-bound procedure in that case.

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.

Acknowledgements Open access funding provided by MTA Institute for Computer Science and Control (MTA SZTAKI).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

8 Appendix

8.1 Proof of Lemma 2

Proof (statement (i)) Let $p, q \in \{1, \dots, n\}$ be distinct elements, $1 \leq j_1 < j_2 < j_3 < j_4 \leq 2n$ and consider points $P_1 = (s^1, e^1), P_2 = (s^2, e^2) \in P_{2n}^{2-chains}$ 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.,

$$s_{p,j_1}^1 = e_{p,j_2}^1 = s_{q,j_3}^1 = e_{q,j_4}^1 = 1 \quad \text{and} \quad s_{p,j_1}^2 = e_{p,j_3}^2 = s_{q,j_2}^2 = e_{q,j_4}^2 = 1,$$

and $s_{r,j}^1 = s_{r,j}^2, e_{r,j}^1 = e_{r,j}^2$ for all $r \notin \{p, q\}$ and $j \in \{1, \dots, 2n\}$. Since P_1 and P_2 satisfy (25), we have

$$\alpha_{p,j_1} + \beta_{p,j_2} + \alpha_{q,j_3} + \beta_{q,j_4} + \sum_{\substack{r=1 \\ r \neq p,q}}^n \sum_{j=1}^{2n} (\alpha_{r,j} s_{r,j}^1 + \beta_{r,j} e_{r,j}^1) = \gamma,$$

and

$$\alpha_{p,j_1} + \beta_{p,j_3} + \alpha_{q,j_2} + \beta_{q,j_4} + \sum_{\substack{r=1 \\ r \neq p,q}}^n \sum_{j=1}^{2n} (\alpha_{r,j} s_{r,j}^2 + \beta_{r,j} e_{r,j}^2) = \gamma,$$

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 $p, q, r \in \{1, \dots, n\}$, 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 $p, q \in \{1, \dots, n\}$ be distinct elements, $1 \leq j_1 < j_2 < j_3 < j_4 \leq 2n$ and consider points $P_1, P_2 \in P_{2n}^{2-chains}$ such that $\sigma_p(P_1) = (j_1, j_3), \sigma_q(P_1) = (j_2, j_4)$

Table 6 Detailed computational results for Family 1 with $n = 50$

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
1	BnB	2524.0	2531.0	2531.0	0.0	12,377	0	54.1
	BnC (Default)	2526.2	2531.0	2531.0	0.0	535	0	13.2
	BnC (Parity)-1	2524.0	2531.0	2531.0	0.0	4	27	1.2
	BnC (Parity)-2	2530.5	2531.0	2531.0	0.0	3	19	0.8
2	BnB	2522.3	2526.0	2526.0	0.0	520	0	6.8
	BnC (Default)	2523.3	2526.0	2526.0	0.0	451	0	4.5
	BnC (Parity)-1	2522.3	2526.0	2526.0	0.0	3	24	0.5
	BnC (Parity)-2	2526.0	2526.0	2526.0	0.0	0	20	0.2
3	BnB	2527.2	2530.0	2530.0	0.0	332	0	6.1
	BnC (Default)	2528.0	2530.0	2530.0	0.0	208	0	2.8
	BnC (Parity)-1	2527.2	2530.0	2530.0	0.0	4	17	0.8
	BnC (Parity)-2	2530.0	2530.0	2530.0	0.0	0	27	0.2
4	BnB	2518.6	2523.0	2523.0	0.0	5693	0	27.5
	BnC (Default)	2519.2	2523.0	2523.0	0.0	2066	0	11.5
	BnC (Parity)-1	2518.6	2523.0	2523.0	0.0	8	22	2.0
	BnC (Parity)-2	2521.7	2523.0	2523.0	0.0	7	23	1.8
5	BnB	2516.7	2519.0	2519.0	0.0	7	0	1.6
	BnC (Default)	2519.0	2519.0	2519.0	0.0	0	0	0.8
	BnC (Parity)-1	2516.7	2519.0	2519.0	0.0	3	17	0.7
	BnC (Parity)-2	2519.0	2519.0	2519.0	0.0	0	23	0.2
6	BnB	2521.0	2525.0	2525.0	0.0	232	0	5.6
	BnC (Default)	2521.8	2525.0	2525.0	0.0	339	0	3.8
	BnC (Parity)-1	2521.0	2525.0	2525.0	0.0	4	17	0.9
	BnC (Parity)-2	2524.5	2525.0	2525.0	0.0	2	18	0.4
7	BnB	2524.6	2527.0	2527.0	0.0	133	0	3.1
	BnC (Default)	2525.6	2527.0	2527.0	0.0	14	0	0.9
	BnC (Parity)-1	2524.6	2527.0	2527.0	0.0	5	12	1.2
	BnC (Parity)-2	2526.2	2527.0	2527.0	0.0	3	15	0.7
8	BnB	2519.0	2523.0	2523.0	0.0	1469	0	12.2
	BnC (Default)	2519.8	2523.0	2523.0	0.0	624	0	4.7
	BnC (Parity)-1	2519.0	2523.0	2523.0	0.0	5	21	1.3
	BnC (Parity)-2	2522.0	2523.0	2523.0	0.0	5	18	0.9
9	BnB	2522.1	2528.0	2528.0	0.0	8774	0	38.1
	BnC (Default)	2522.3	2528.0	2528.0	0.0	3460	0	29.6
	BnC (Parity)-1	2522.1	2528.0	2528.0	0.0	4	18	1.2
	BnC (Parity)-2	2528.0	2528.0	2528.0	0.0	0	20	0.2
10	BnB	2515.9	2522.0	2522.0	0.0	2424	0	17.0
	BnC (Default)	2517.9	2522.0	2522.0	0.0	702	0	5.4
	BnC (Parity)-1	2515.9	2522.0	2522.0	0.0	3	22	0.9
	BnC (Parity)-2	2521.8	2522.0	2522.0	0.0	3	19	0.5

Table 6 continued

Instance	Method	LB ^f	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
Avg	BnB	2521.1	2525.4	2525.4	0.0	3196.1	0.0	17.2
	BnC (Default)	2522.3	2525.4	2525.4	0.0	839.9	0.0	7.7
	BnC (Parity)-1	2521.1	2525.4	2525.4	0.0	4.3	19.7	1.1
	BnC (Parity)-2	2525.0	2525.4	2525.4	0.0	2.3	20.2	0.6

Table 7 Detailed computational results for Family 1 with $n = 100$

Instance	Method	LB ^f	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
1	BnB	5004.7	5006.7	5016.0	0.2	19,580	0	600.0
	BnC (Default)	5004.8	5009.0	5009.0	0.0	18,317	0	513.5
	BnC (Parity)-1	5004.7	5009.0	5009.0	0.0	160	40	25.0
	BnC (Parity)-2	5007.3	5009.0	5009.0	0.0	26	28	15.2
2	BnB	5005.0	5007.0	5007.0	0.0	3850	0	148.7
	BnC (Default)	5005.1	5007.0	5007.0	0.0	2842	0	77.6
	BnC (Parity)-1	5005.0	5007.0	5007.0	0.0	69	32	22.4
	BnC (Parity)-2	5005.9	5007.0	5007.0	0.0	112	29	24.6
3	BnB	5004.7	5008.0	5008.0	0.0	9963	0	282.5
	BnC (Default)	5004.9	5008.0	5008.0	0.0	4339	0	177.8
	BnC (Parity)-1	5004.7	5008.0	5008.0	0.0	171	36	22.9
	BnC (Parity)-2	5006.1	5008.0	5008.0	0.0	130	32	25.0
4	BnB	5007.4	5008.7	5058.0	1.0	21,653	0	600.0
	BnC (Default)	5007.6	5008.5	5021.0	0.3	10,395	0	600.0
	BnC (Parity)-1	5007.4	5013.0	5013.0	0.0	102	38	21.4
	BnC (Parity)-2	5011.7	5013.0	5013.0	0.0	81	22	20.4
5	BnB	5003.9	5006.0	5011.0	0.1	18,585	0	600.0
	BnC (Default)	5004.7	5008.0	5008.0	0.0	16,913	0	524.5
	BnC (Parity)-1	5003.9	5008.0	5008.0	0.0	263	39	37.0
	BnC (Parity)-2	5005.8	5008.0	5008.0	0.0	320	39	38.3
6	BnB	5006.9	5010.0	5010.0	0.0	5414	0	217.9
	BnC (Default)	5007.1	5010.0	5010.0	0.0	3986	0	212.0
	BnC (Parity)-1	5006.9	5010.0	5010.0	0.0	62	26	18.4
	BnC (Parity)-2	5008.4	5010.0	5010.0	0.0	81	24	17.4
7	BnB	5004.0	5005.6	5010.0	0.1	21,114	0	600.0
	BnC (Default)	5004.2	5007.0	5007.0	0.0	21,630	0	483.0
	BnC (Parity)-1	5004.0	5007.0	5007.0	0.0	221	49	29.3
	BnC (Parity)-2	5005.7	5007.0	5007.0	0.0	168	25	24.9
8	BnB	5004.1	5007.0	5007.0	0.0	1633	0	66.5
	BnC (Default)	5004.5	5007.0	5007.0	0.0	11,646	0	517.4
	BnC (Parity)-1	5004.1	5007.0	5007.0	0.0	122	19	23.5
	BnC (Parity)-2	5005.7	5007.0	5007.0	0.0	112	20	27.7

Table 7 continued

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
9	BnB	5004.3	5005.6	5087.0	1.6	22,484	0	600.0
	BnC (Default)	5005.1	5005.9	5028.0	0.4	11,269	0	600.0
	BnC (Parity)-1	5004.3	5011.0	5011.0	0.0	138	41	24.7
	BnC (Parity)-2	5008.7	5011.0	5011.0	0.0	235	27	26.7
10	BnB	5002.6	5005.0	5005.0	0.0	15,896	0	453.6
	BnC (Default)	5002.8	5005.0	5005.0	0.0	5367	0	266.3
	BnC (Parity)-1	5002.6	5005.0	5005.0	0.0	92	37	25.6
	BnC (Parity)-2	5004.0	5005.0	5005.0	0.0	11	18	12.6
Avg	BnB	5004.8	5006.9	5021.9	0.3	14,017.2	0.0	416.9
	BnC (Default)	5005.1	5007.5	5011.0	0.1	10,670.4	0.0	397.2
	BnC (Parity)-1	5004.8	5008.5	5008.5	0.0	140.0	35.7	25.0
	BnC (Parity)-2	5006.9	5008.5	5008.5	0.0	127.6	26.4	23.3

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 (25), 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 $P_3, P_4 \in P_{2n}^{2-chains}$ 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 (25), 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 $p, q \in \{1, \dots, n\}$ be distinct elements, $1 \leq j_1 < j_2 < j_3 < j_4 \leq 2n$ and consider points $P_1, P_2 \in P_{2n}^{2-chains}$ 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 (25), 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 $P_3, P_4 \in P_{2n}^{2-chains}$ 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 (25), 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 $p, q \in \{1, \dots, t\}$ be distinct elements, $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$ and consider points $P_1, P_2 \in P_{2n}^{parity}$ 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.,

$$s_{p,j_1}^1 = e_{p,j_3}^1 = s_{q,j_2}^1 = e_{q,j_4}^1 = 1 \quad \text{and} \quad s_{p,j_2}^2 = e_{p,j_3}^2 = s_{q,j_1}^2 = e_{q,j_4}^2 = 1,$$

Table 8 Detailed computational results for Family 1 with $n = 150$

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
1	BnB	7500.0	7500.0	7529.0	0.4	7205	0	600.0
	BnC (Default)	7500.0	7500.0	7500.0	0.0	472	0	228.2
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	6	26	31.9
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	52	17	100.4
2	BnB	7500.0	7500.0	7500.0	0.0	1054	0	193.7
	BnC (Default)	7500.0	7500.0	7500.0	0.0	0	0	31.0
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	9	31	38.6
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	18	20	58.4
3	BnB	7500.0	7500.0	7500.0	0.0	5791	0	421.3
	BnC (Default)	7500.0	7500.0	7500.0	0.0	135	0	88.5
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	13	35	50.2
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	84	28	88.6
4	BnB	7500.0	7500.0	7500.0	0.0	847	0	200.3
	BnC (Default)	7500.0	7500.0	7500.0	0.0	2753	0	243.7
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	10	40	45.1
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	378	48	113.2
5	BnB	7500.0	7500.0	7500.0	0.0	42	0	70.2
	BnC (Default)	7500.0	7500.0	7500.0	0.0	2461	0	495.3
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	16	37	53.0
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	150	40	106.9
6	BnB	7500.0	7500.0	7604.0	1.4	5132	0	600.0
	BnC (Default)	7500.0	7500.0	7501.0	0.0	4221	0	600.0
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	4	78	29.2
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	33	27	80.2
7	BnB	7500.0	7500.0	7500.0	0.0	413	0	114.7
	BnC (Default)	7500.0	7500.0	7500.0	0.0	0	0	18.6
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	2	26	18.5
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	24	29	71.1
8	BnB	7500.0	7500.0	7500.0	0.0	41	0	65.4
	BnC (Default)	7500.0	7500.0	7500.0	0.0	246	0	89.3
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	9	26	38.8
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	54	21	84.3
9	BnB	7500.0	7500.0	7502.0	0.0	7464	0	600.0
	BnC (Default)	7500.0	7500.0	7500.0	0.0	785	0	129.3
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	9	27	41.5
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	3	9	34.0
10	BnB	7500.0	7500.0	7502.0	0.0	9417	0	600.0
	BnC (Default)	7500.0	7500.0	7500.0	0.0	1499	0	351.0
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	50	28	81.3
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	32	28	83.2

Table 8 continued

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
Avg	BnB	7500.0	7500.0	7513.7	0.2	3740.6	0.0	346.6
	BnC (Default)	7500.0	7500.0	7500.1	0.0	1257.2	0.0	227.5
	BnC (Parity)-1	7500.0	7500.0	7500.0	0.0	12.8	35.4	42.8
	BnC (Parity)-2	7500.0	7500.0	7500.0	0.0	82.8	26.7	82.0

and $s_{r,j}^1 = s_{r,j}^2, e_{r,j}^1 = e_{r,j}^2$ for all $r \notin \{p, q\}$ and $j \in \{1, \dots, 2n\}$. Note that such points exist according to Remark 2. Since P_1 and P_2 satisfy (31), we have

$$\alpha_{p,j_1} + \beta_{p,j_3} + \alpha_{q,j_2} + \beta_{q,j_4} + \sum_{\substack{r=1 \\ r \neq p,q}}^n \sum_{j=1}^{2n} (\alpha_{r,j} s_{r,j}^1 + \beta_{r,j} e_{r,j}^1) = \gamma,$$

and

$$\alpha_{p,j_2} + \beta_{p,j_3} + \alpha_{q,j_1} + \beta_{q,j_4} + \sum_{\substack{r=1 \\ r \neq p,q}}^n \sum_{j=1}^{2n} (\alpha_{r,j} s_{r,j}^2 + \beta_{r,j} e_{r,j}^2) = \gamma,$$

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 $p, q \in \{1, \dots, t\}$ be distinct elements, $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$ and consider points $P_1, P_2 \in P_{2n}^{parity}$ 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 (31) we have $\beta_{p,j_4} - \beta_{p,j_3} = \beta_{q,j_4} - \beta_{q,j_3}$.

Proof (statement (ii)) Let $p, q \in \{1, \dots, t\}$ be distinct elements and $1 \leq j_1 \leq 2k < j_2 < j_3 < j_4 \leq 2n$. First, consider points $P_1, P_2 \in P_{2n}^{parity}$ 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 (31) 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 $P_3, P_4 \in P_{2n}^{parity}$ 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 $\sigma_r(P_3) = \sigma_r(P_4)$ for all $r \notin \{p, q\}$. Since P_3 and P_4 satisfy (31) 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 $p, q \in \{1, \dots, t\}$ be distinct elements, $1 \leq j_1 < j_2 < j_3 \leq 2k < j_4 \leq 2n$. First, consider points $P_1, P_2 \in P_{2n}^{parity}$ such that $\sigma_p(P_1) = (j_1, j_3), \sigma_q(P_1) = (j_2, j_4)$ and $\sigma_q(P_2) = (j_1, j_4), \sigma_p(P_2) = (j_2, j_3)$ and $\sigma_r(P_1) = \sigma_r(P_2)$ for all $r \notin \{p, q\}$. Since P_1 and P_2 satisfy (31) 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 $P_3, P_4 \in P_{2n}^{parity}$ 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 (31) 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 $j' = 2$. □

Table 9 Detailed computational results for Family 2 with $n = 50$

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
1	BnB	1797.5	1816.0	1816.0	0.0	1978	0	5.4
	BnC (Default)	1816.0	1816.0	1816.0	0.0	0	0	0.4
	BnC (Parity)-1	1797.5	1816.0	1816.0	0.0	5	2	0.5
	BnC (Parity)-2	1815.7	1816.0	1816.0	0.0	3	3	0.4
2	BnB	1804.5	1819.0	1819.0	0.0	495	0	2.2
	BnC (Default)	1815.0	1819.0	1819.0	0.0	28	0	0.8
	BnC (Parity)-1	1804.5	1819.0	1819.0	0.0	3	3	0.3
	BnC (Parity)-2	1819.0	1819.0	1819.0	0.0	0	2	0.2
3	BnB	1798.5	1817.0	1817.0	0.0	751	0	2.3
	BnC (Default)	1817.0	1817.0	1817.0	0.0	0	0	0.3
	BnC (Parity)-1	1798.5	1817.0	1817.0	0.0	4	4	0.2
	BnC (Parity)-2	1816.5	1817.0	1817.0	0.0	3	4	0.1
4	BnB	1798.8	1814.0	1814.0	0.0	530	0	2.8
	BnC (Default)	1814.0	1814.0	1814.0	0.0	0	0	0.8
	BnC (Parity)-1	1798.8	1814.0	1814.0	0.0	4	3	0.3
	BnC (Parity)-2	1814.0	1814.0	1814.0	0.0	2	3	0.2
5	BnB	1799.0	1817.0	1817.0	0.0	845	0	3.5
	BnC (Default)	1817.0	1817.0	1817.0	0.0	0	0	0.5
	BnC (Parity)-1	1799.0	1817.0	1817.0	0.0	3	2	0.3
	BnC (Parity)-2	1817.0	1817.0	1817.0	0.0	0	2	0.2
6	BnB	1806.0	1822.0	1822.0	0.0	1835	0	5.3
	BnC (Default)	1822.0	1822.0	1822.0	0.0	0	0	0.6
	BnC (Parity)-1	1806.0	1822.0	1822.0	0.0	4	2	0.4
	BnC (Parity)-2	1821.3	1822.0	1822.0	0.0	4	2	0.5
7	BnB	1800.5	1817.0	1817.0	0.0	670	0	3.0
	BnC (Default)	1814.8	1817.0	1817.0	0.0	119	0	1.2
	BnC (Parity)-1	1800.5	1817.0	1817.0	0.0	3	2	0.2
	BnC (Parity)-2	1817.0	1817.0	1817.0	0.0	0	3	0.1
8	BnB	1799.0	1814.0	1814.0	0.0	799	0	2.8
	BnC (Default)	1814.0	1814.0	1814.0	0.0	0	0	0.5
	BnC (Parity)-1	1799.0	1814.0	1814.0	0.0	3	3	0.2
	BnC (Parity)-2	1814.0	1814.0	1814.0	0.0	2	2	0.2
9	BnB	1803.5	1819.0	1819.0	0.0	533	0	2.5
	BnC (Default)	1819.0	1819.0	1819.0	0.0	0	0	0.3
	BnC (Parity)-1	1803.5	1819.0	1819.0	0.0	3	4	0.2
	BnC (Parity)-2	1819.0	1819.0	1819.0	0.0	0	3	0.1
10	BnB	1797.0	1810.0	1810.0	0.0	815	0	2.8
	BnC (Default)	1810.0	1810.0	1810.0	0.0	0	0	0.5
	BnC (Parity)-1	1797.0	1810.0	1810.0	0.0	3	2	0.2
	BnC (Parity)-2	1810.0	1810.0	1810.0	0.0	3	2	0.1

Table 9 continued

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
Avg	BnB	1800.4	1816.5	1816.5	0.0	925.1	0.0	3.3
	BnC (Default)	1815.9	1816.5	1816.5	0.0	14.7	0.0	0.6
	BnC (Parity)-1	1800.4	1816.5	1816.5	0.0	3.5	2.7	0.3
	BnC (Parity)-2	1816.3	1816.5	1816.5	0.0	1.7	2.6	0.2

Table 10 Detailed computational results for Family 2 with $n = 100$

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
1	BnB	3595.1	3598.5	3650.0	1.4	58,120	0	600.0
	BnC (Default)	3595.5	3613.5	3644.0	0.8	15,623	0	600.0
	BnC (Parity)-1	3595.1	3642.0	3642.0	0.0	3	24	3.4
	BnC (Parity)-2	3642.0	3642.0	3642.0	0.0	3	24	3.7
2	BnB	3595.0	3598.3	3652.0	1.5	75,054	0	600.0
	BnC (Default)	3600.3	3613.2	3642.0	0.8	15,739	0	600.0
	BnC (Parity)-1	3595.0	3641.0	3641.0	0.0	4	27	4.9
	BnC (Parity)-2	3641.0	3641.0	3641.0	0.0	0	23	3.0
3	BnB	3599.0	3599.5	3670.0	1.9	57,020	0	600.0
	BnC (Default)	3608.5	3617.8	3645.0	0.8	14,084	0	600.0
	BnC (Parity)-1	3599.0	3643.0	3643.0	0.0	4	26	5.6
	BnC (Parity)-2	3642.5	3643.0	3643.0	0.0	2	16	4.6
4	BnB	3594.5	3599.5	3655.0	1.5	73,846	0	600.0
	BnC (Default)	3601.0	3617.3	3644.0	0.7	14,320	0	600.0
	BnC (Parity)-1	3594.5	3642.0	3642.0	0.0	3	29	4.2
	BnC (Parity)-2	3642.0	3642.0	3642.0	0.0	0	22	2.7
5	BnB	3594.5	3594.5	3661.0	1.8	77,634	0	600.0
	BnC (Default)	3599.0	3611.0	3643.0	0.9	14,053	0	600.0
	BnC (Parity)-1	3594.5	3640.0	3640.0	0.0	7	31	7.1
	BnC (Parity)-2	3640.0	3640.0	3640.0	0.0	4	8	5.9
6	BnB	3600.8	3603.8	3673.0	1.9	52,608	0	600.0
	BnC (Default)	3602.0	3614.0	3649.0	1.0	16,714	0	600.0
	BnC (Parity)-1	3600.8	3646.0	3646.0	0.0	3	24	4.2
	BnC (Parity)-2	3646.0	3646.0	3646.0	0.0	2	22	3.3
7	BnB	3594.5	3596.5	3663.0	1.8	54,950	0	600.0
	BnC (Default)	3604.8	3624.8	3642.0	0.5	11,172	0	600.0
	BnC (Parity)-1	3594.5	3641.0	3641.0	0.0	3	18	4.6
	BnC (Parity)-2	3641.0	3641.0	3641.0	0.0	10	12	9.9
8	BnB	3597.0	3597.0	3667.0	1.9	53,452	0	600.0
	BnC (Default)	3609.0	3628.5	3647.0	0.5	13,020	0	600.0
	BnC (Parity)-1	3597.0	3643.0	3643.0	0.0	4	27	4.0
	BnC (Parity)-2	3643.0	3643.0	3643.0	0.0	2	17	3.1

Table 10 continued

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
9	BnB	3597.5	3598.5	3691.0	2.5	75,310	0	600.0
	BnC (Default)	3602.0	3617.3	3646.0	0.8	14,620	0	600.0
	BnC (Parity)-1	3597.5	3642.0	3642.0	0.0	4	19	5.4
	BnC (Parity)-2	3642.0	3642.0	3642.0	0.0	2	11	3.9
10	BnB	3594.5	3599.5	3652.0	1.4	59,084	0	600.0
	BnC (Default)	3600.5	3607.5	3643.0	1.0	18,283	0	600.0
	BnC (Parity)-1	3594.5	3641.0	3641.0	0.0	8	46	10.4
	BnC (Parity)-2	3641.0	3641.0	3641.0	0.0	20	14	11.3
Avg	BnB	3596.2	3598.6	3663.4	1.8	63,707.8	0.0	600.0
	BnC (Default)	3602.3	3616.5	3644.5	0.8	14,762.8	0.0	600.0
	BnC (Parity)-1	3596.2	3642.1	3642.1	0.0	4.3	27.1	5.4
	BnC (Parity)-2	3642.1	3642.1	3642.1	0.0	4.5	16.9	5.2

Table 11 Detailed computational results for Family 2 with $n = 150$

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
1	BnB	5342.0	5342.3	5392.0	0.9	9215	0	600.0
	BnC (Default)	5346.0	5357.0	5362.0	0.1	4631	0	600.0
	BnC (Parity)-1	5342.0	5362.0	5362.0	0.0	5	6	14.1
	BnC (Parity)-2	5362.0	5362.0	5362.0	0.0	3	4	11.0
2	BnB	5340.0	5340.0	5392.0	1.0	10,210	0	600.0
	BnC (Default)	5343.5	5347.0	5360.0	0.2	8701	0	600.0
	BnC (Parity)-1	5340.0	5360.0	5360.0	0.0	4	3	14.6
	BnC (Parity)-2	5360.0	5360.0	5360.0	0.0	3	7	12.7
3	BnB	5340.5	5341.0	5386.0	0.8	9459	0	600.0
	BnC (Default)	5345.5	5354.0	5360.0	0.1	5531	0	600.0
	BnC (Parity)-1	5340.5	5360.0	5360.0	0.0	8	7	20.7
	BnC (Parity)-2	5360.0	5360.0	5360.0	0.0	16	6	27.9
4	BnB	5341.0	5341.0	5478.0	2.5	10,611	0	600.0
	BnC (Default)	5342.5	5358.0	5361.0	0.1	7004	0	600.0
	BnC (Parity)-1	5341.0	5361.0	5361.0	0.0	5	2	13.8
	BnC (Parity)-2	5361.0	5361.0	5361.0	0.0	5	23	15.1
5	BnB	5342.0	5342.0	5404.0	1.2	10,160	0	600.0
	BnC (Default)	5347.3	5361.0	5361.0	0.0	2253	0	340.2
	BnC (Parity)-1	5342.0	5361.0	5361.0	0.0	3	10	12.0
	BnC (Parity)-2	5361.0	5361.0	5361.0	0.0	3	5	11.6
6	BnB	5340.0	5340.3	5384.0	0.8	8241	0	600.0
	BnC (Default)	5342.3	5347.8	5360.0	0.2	3865	0	600.0
	BnC (Parity)-1	5340.0	5360.0	5360.0	0.0	5	9	17.2
	BnC (Parity)-2	5360.0	5360.0	5360.0	0.0	7	13	20.0

Table 11 continued

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
7	BnB	5341.0	5341.0	5398.0	1.1	8838	0	600.0
	BnC (Default)	5343.5	5349.5	5361.0	0.2	5235	0	600.0
	BnC (Parity)-1	5341.0	5360.0	5360.0	0.0	4	7	11.3
	BnC (Parity)-2	5360.0	5360.0	5360.0	0.0	11	8	22.9
8	BnB	5340.0	5340.0	5386.0	0.9	9313	0	600.0
	BnC (Default)	5345.0	5350.5	5360.0	0.2	3570	0	600.0
	BnC (Parity)-1	5340.0	5360.0	5360.0	0.0	7	6	18.8
	BnC (Parity)-2	5360.0	5360.0	5360.0	0.0	13	22	26.2
9	BnB	5340.0	5340.5	5377.0	0.7	7623	0	600.0
	BnC (Default)	5342.0	5350.0	5360.0	0.2	3672	0	600.0
	BnC (Parity)-1	5340.0	5360.0	5360.0	0.0	3	5	14.6
	BnC (Parity)-2	5360.0	5360.0	5360.0	0.0	3	3	16.8
10	BnB	5340.5	5341.0	5401.0	1.1	7325	0	600.0
	BnC (Default)	5345.3	5353.2	5360.0	0.1	4987	0	600.0
	BnC (Parity)-1	5340.5	5360.0	5360.0	0.0	6	14	16.5
	BnC (Parity)-2	5360.0	5360.0	5360.0	0.0	4	9	17.4
Avg	BnB	5340.7	5340.9	5399.8	1.1	9099.5	0.0	600.0
	BnC (Default)	5344.3	5352.8	5360.5	0.1	4944.9	0.0	574.0
	BnC (Parity)-1	5340.7	5360.4	5360.4	0.0	5.0	6.9	15.4
	BnC (Parity)-2	5360.4	5360.4	5360.4	0.0	6.8	10.0	18.1

Table 12 Detailed computational results for Family 3 with $n = 50$

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
1	BnB	2133.7	2152.0	2152.0	0.0	41,718	0	77.2
	BnC (Default)	2145.4	2152.0	2152.0	0.0	1475	0	3.2
	BnC (Parity)-1	2133.7	2152.0	2152.0	0.0	4	6	0.4
	BnC (Parity)-2	2151.5	2152.0	2152.0	0.0	3	5	0.3
2	BnB	1986.8	2003.0	2003.0	0.0	3628	0	4.3
	BnC (Default)	1995.8	2003.0	2003.0	0.0	1178	0	1.6
	BnC (Parity)-1	1986.8	2003.0	2003.0	0.0	4	3	0.1
	BnC (Parity)-2	2002.8	2003.0	2003.0	0.0	3	5	0.1
3	BnB	2032.0	2048.0	2048.0	0.0	13,765	0	21.2
	BnC (Default)	2036.9	2048.0	2048.0	0.0	534	0	2.9
	BnC (Parity)-1	2032.0	2048.0	2048.0	0.0	3	11	0.2
	BnC (Parity)-2	2048.0	2048.0	2048.0	0.0	0	6	0.1

Table 12 continued

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
4	BnB	2098.0	2114.0	2114.0	0.0	15,020	0	29.8
	BnC (Default)	2109.3	2114.0	2114.0	0.0	1217	0	4.0
	BnC (Parity)-1	2098.0	2114.0	2114.0	0.0	3	9	0.3
	BnC (Parity)-2	2113.8	2114.0	2114.0	0.0	3	7	0.3
5	BnB	2029.0	2046.0	2046.0	0.0	32,895	0	52.2
	BnC (Default)	2039.7	2046.0	2046.0	0.0	224	0	1.1
	BnC (Parity)-1	2029.0	2046.0	2046.0	0.0	6	4	0.3
	BnC (Parity)-2	2044.9	2046.0	2046.0	0.0	4	4	0.4
6	BnB	2090.2	2108.0	2108.0	0.0	36,446	0	66.7
	BnC (Default)	2103.8	2108.0	2108.0	0.0	282	0	1.4
	BnC (Parity)-1	2090.2	2108.0	2108.0	0.0	8	13	0.5
	BnC (Parity)-2	2106.9	2108.0	2108.0	0.0	6	8	0.4
7	BnB	2056.2	2074.0	2074.0	0.0	99,392	0	163.1
	BnC (Default)	2070.0	2074.0	2074.0	0.0	120	0	0.7
	BnC (Parity)-1	2056.2	2074.0	2074.0	0.0	4	9	0.7
	BnC (Parity)-2	2072.4	2074.0	2074.0	0.0	5	5	0.7
8	BnB	1986.5	2001.0	2001.0	0.0	30,937	0	36.0
	BnC (Default)	1996.6	2001.0	2001.0	0.0	347	0	1.0
	BnC (Parity)-1	1986.5	2001.0	2001.0	0.0	4	6	0.2
	BnC (Parity)-2	2000.9	2001.0	2001.0	0.0	2	5	0.2
9	BnB	1982.5	1997.0	1997.0	0.0	8791	0	9.9
	BnC (Default)	1987.6	1997.0	1997.0	0.0	2630	0	2.9
	BnC (Parity)-1	1982.5	1997.0	1997.0	0.0	3	8	0.1
	BnC (Parity)-2	1997.0	1997.0	1997.0	0.0	0	4	0.1
10	BnB	2001.1	2018.0	2018.0	0.0	6334	0	9.9
	BnC (Default)	2012.6	2018.0	2018.0	0.0	504	0	1.2
	BnC (Parity)-1	2001.1	2018.0	2018.0	0.0	3	9	0.2
	BnC (Parity)-2	2017.5	2018.0	2018.0	0.0	3	4	0.1
Avg	BnB	2039.6	2056.1	2056.1	0.0	28,892.6	0.0	47.0
	BnC (Default)	2049.8	2056.1	2056.1	0.0	851.1	0.0	2.0
	BnC (Parity)-1	2039.6	2056.1	2056.1	0.0	4.2	7.8	0.3
	BnC (Parity)-2	2055.6	2056.1	2056.1	0.0	2.9	5.3	0.3

Proof (statement (v)) Let $p, q \in \{1, \dots, t\}$ be distinct elements, $1 \leq j_1 < j_2 < j_3 \leq 2k < j_4 \leq 2n$ and consider points $P_1, P_2 \in P_{2n}^{parity}$ 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 (31) 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 $p, q, r \in \{1, \dots, t\}$, 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 $p = q$. □

Table 13 Detailed computational results for Family 3 with $n = 100$

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
1	BnB	4156.5	4158.6	4203.0	1.1	49,005	0	600.0
	BnC (Default)	4160.1	4174.8	4188.0	0.3	13,032	0	600.0
	BnC (Parity)-1	4156.5	4184.0	4184.0	0.0	17	16	8.0
	BnC (Parity)-2	4184.0	4184.0	4184.0	0.0	24	5	7.8
2	BnB	3995.3	3998.4	4018.0	0.5	57,409	0	600.0
	BnC (Default)	3995.5	4001.9	4014.0	0.3	28,323	0	600.0
	BnC (Parity)-1	3995.3	4013.0	4013.0	0.0	34	14	7.8
	BnC (Parity)-2	4012.2	4013.0	4013.0	0.0	20	6	5.0
3	BnB	4164.2	4166.1	4216.0	1.2	46,005	0	600.0
	BnC (Default)	4172.0	4188.0	4195.0	0.2	12,674	0	600.0
	BnC (Parity)-1	4164.2	4192.0	4192.0	0.0	87	7	11.3
	BnC (Parity)-2	4191.0	4192.0	4192.0	0.0	214	19	14.9
4	BnB	4200.5	4202.5	4249.0	1.1	52,099	0	600.0
	BnC (Default)	4204.0	4213.6	4234.0	0.5	13,490	0	600.0
	BnC (Parity)-1	4200.5	4229.0	4229.0	0.0	55	23	14.9
	BnC (Parity)-2	4227.5	4229.0	4229.0	0.0	21	12	7.7
5	BnB	3995.3	3997.8	4021.0	0.6	68,900	0	600.0
	BnC (Default)	3995.5	4010.9	4015.0	0.1	29,273	0	600.0
	BnC (Parity)-1	3995.3	4014.0	4014.0	0.0	43	8	7.6
	BnC (Parity)-2	4012.9	4014.0	4014.0	0.0	35	3	6.6
6	BnB	4044.5	4047.3	4077.0	0.7	57,242	0	600.0
	BnC (Default)	4049.5	4072.0	4072.0	0.0	18,567	0	551.0
	BnC (Parity)-1	4044.5	4072.0	4072.0	0.0	58	17	8.7
	BnC (Parity)-2	4070.7	4072.0	4072.0	0.0	28	9	5.7
7	BnB	4082.7	4085.3	4129.0	1.1	59,258	0	600.0
	BnC (Default)	4087.5	4100.9	4116.0	0.4	12,009	0	600.0
	BnC (Parity)-1	4082.7	4111.0	4111.0	0.0	173	24	12.5
	BnC (Parity)-2	4109.2	4111.0	4111.0	0.0	22	8	6.4
8	BnB	3993.5	3995.9	4014.0	0.5	75,567	0	600.0
	BnC (Default)	3997.9	4009.8	4012.0	0.1	45,137	0	600.0
	BnC (Parity)-1	3993.5	4012.0	4012.0	0.0	80	21	8.3
	BnC (Parity)-2	4010.8	4012.0	4012.0	0.0	63	12	8.4
9	BnB	3952.7	3955.9	3975.0	0.5	66,204	0	600.0
	BnC (Default)	3954.8	3971.0	3971.0	0.0	50,761	0	551.7
	BnC (Parity)-1	3952.7	3971.0	3971.0	0.0	6	9	3.4
	BnC (Parity)-2	3970.3	3971.0	3971.0	0.0	5	5	3.1
10	BnB	3950.8	3953.5	3975.0	0.5	71,367	0	600.0
	BnC (Default)	3950.8	3961.5	3970.0	0.2	46,277	0	600.0
	BnC (Parity)-1	3950.8	3969.0	3969.0	0.0	7	21	3.9
	BnC (Parity)-2	3968.1	3969.0	3969.0	0.0	12	6	3.4

Table 13 continued

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
Avg	BnB	4053.6	4056.1	4087.7	0.8	60,305.6	0.0	600.0
	BnC (Default)	4056.8	4070.4	4078.7	0.2	26,954.3	0.0	590.3
	BnC (Parity)-1	4053.6	4076.7	4076.7	0.0	56.0	16.0	8.6
	BnC (Parity)-2	4075.7	4076.7	4076.7	0.0	44.4	8.5	6.9

Proof (statement (vi)) Let $p, q \in \{1, \dots, t\}$ be distinct elements, $1 \leq j_1 \leq 2k < j_2 < j_3 < j_4 \leq 2n$ and consider points $P_1, P_2 \in P_{2n}^{parity}$ 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 (31) 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 $p, q, r \in \{1, \dots, t\}$, 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 $p = q$. □

8.3 Proof of Lemma 4

Proof (statement (vii)) Let $p \in \{1, \dots, t\}, \bar{q} \in \{t + 1, \dots, n\}$ and $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_{2n}^{parity}$ 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 (31) we have $\alpha_{p,j_2} - \alpha_{p,j_1} = \alpha_{\bar{q},j_2} - \alpha_{\bar{q},j_1}$. □

Proof (statement (viii)) Let $p \in \{1, \dots, t\}, \bar{q} \in \{t + 1, \dots, n\}$ and $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_{2n}^{parity}$ 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 (31) we have $\beta_{p,j_4} - \beta_{p,j_3} = \beta_{\bar{q},j_4} - \beta_{\bar{q},j_3}$. □

Proof (statement (ix)) Let $p \in \{1, \dots, t\}, \bar{q} \in \{t + 1, \dots, n\}$ and $1 \leq j_1 < j_2 < j_3 \leq 2k < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_{2n}^{parity}$ 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 (31) we have $\alpha_{p,j_3} - \alpha_{p,j_2} = \beta_{\bar{q},j_3} - \beta_{\bar{q},j_2}$. □

Proof (statement (x)) Let $p \in \{1, \dots, t\}, \bar{q} \in \{t + 1, \dots, n\}$ and $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_{2n}^{parity}$ 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 (31) we have $\alpha_{p,j_3} - \alpha_{p,j_2} = \beta_{\bar{q},j_3} - \beta_{\bar{q},j_2}$. □

Proof (statement (xi)) Let $p \in \{1, \dots, t\}, \bar{q} \in \{t + 1, \dots, n\}$ and $1 \leq j_1 < j_2 \leq 2k < j_3 < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_{2n}^{parity}$ 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 (31) we have $\beta_{p,j_3} - \beta_{p,j_2} = \alpha_{\bar{q},j_3} - \alpha_{\bar{q},j_2}$. □

Proof (statement (xii)) Let $p \in \{1, \dots, t\}, \bar{q} \in \{t + 1, \dots, n\}$ and $1 \leq j_1 \leq 2k < j_2 < j_3 < j_4 \leq 2n$. Consider points $P_1, P_2 \in P_{2n}^{parity}$ 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 (31) we have $\beta_{p,j_3} - \beta_{p,j_2} = \alpha_{\bar{q},j_3} - \alpha_{\bar{q},j_2}$. □

Table 14 Detailed computational results for Family 3 with $n = 150$

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
1	BnB	6332.0	6332.0	6399.0	1.1	10,615	0	600.0
	BnC (Default)	6334.5	6337.0	6356.0	0.3	6019	0	600.0
	BnC (Parity)-1	6332.0	6351.0	6351.0	0.0	22	22	38.0
	BnC (Parity)-2	6351.0	6351.0	6351.0	0.0	621	23	88.8
2	BnB	6092.7	6093.0	6146.0	0.9	14,605	0	600.0
	BnC (Default)	6094.9	6101.9	6115.0	0.2	10,180	0	600.0
	BnC (Parity)-1	6092.7	6112.0	6112.0	0.0	17	9	23.6
	BnC (Parity)-2	6111.5	6112.0	6112.0	0.0	421	11	52.4
3	BnB	6254.0	6254.0	6309.0	0.9	11,203	0	600.0
	BnC (Default)	6254.0	6256.0	6277.0	0.3	8483	0	600.0
	BnC (Parity)-1	6254.0	6273.0	6273.0	0.0	7	8	16.4
	BnC (Parity)-2	6273.0	6273.0	6273.0	0.0	655	11	84.4
4	BnB	6171.0	6171.0	6244.0	1.2	13,616	0	600.0
	BnC (Default)	6172.5	6173.5	6194.0	0.3	7869	0	600.0
	BnC (Parity)-1	6171.0	6191.0	6191.0	0.0	3	21	11.2
	BnC (Parity)-2	6191.0	6191.0	6191.0	0.0	28	5	35.2
5	BnB	5954.0	5954.8	5996.0	0.7	18,052	0	600.0
	BnC (Default)	5957.9	5963.3	5975.0	0.2	10,208	0	600.0
	BnC (Parity)-1	5954.0	5973.0	5973.0	0.0	55	14	28.2
	BnC (Parity)-2	5972.1	5973.0	5973.0	0.0	69	6	27.7
6	BnB	6016.2	6017.1	6059.0	0.7	14,655	0	600.0
	BnC (Default)	6016.3	6017.1	6041.0	0.4	10,356	0	600.0
	BnC (Parity)-1	6016.2	6036.0	6036.0	0.0	60	18	27.7
	BnC (Parity)-2	6034.7	6036.0	6036.0	0.0	115	10	26.4
7	BnB	6051.0	6051.3	6087.0	0.6	17,068	0	600.0
	BnC (Default)	6052.0	6057.0	6070.0	0.2	7370	0	600.0
	BnC (Parity)-1	6051.0	6070.0	6070.0	0.0	98	22	39.4
	BnC (Parity)-2	6070.0	6070.0	6070.0	0.0	29	17	29.9
8	BnB	5932.0	5932.2	5963.0	0.5	20,137	0	600.0
	BnC (Default)	5932.0	5933.0	5952.0	0.3	12,154	0	600.0
	BnC (Parity)-1	5932.0	5951.0	5951.0	0.0	4	21	8.2
	BnC (Parity)-2	5950.1	5951.0	5951.0	0.0	12	16	16.9
9	BnB	5891.0	5891.8	5929.0	0.6	22,698	0	600.0
	BnC (Default)	5893.0	5908.8	5911.0	0.0	15,587	0	600.0
	BnC (Parity)-1	5891.0	5911.0	5911.0	0.0	6	10	8.7
	BnC (Parity)-2	5911.0	5911.0	5911.0	0.0	5	5	9.2
10	BnB	5930.1	5930.8	5963.0	0.5	20,668	0	600.0
	BnC (Default)	5930.1	5935.4	5952.0	0.3	11,010	0	600.0
	BnC (Parity)-1	5930.1	5950.0	5950.0	0.0	57	19	32.0
	BnC (Parity)-2	5950.0	5950.0	5950.0	0.0	14	2	12.5

Table 14 continued

Instance	Method	LB ^r	LB ^f	UB ^f	Gap ^f	Nodes	Cuts	Time
Avg	BnB	6062.4	6062.8	6109.5	0.8	16,331.7	0.0	600.0
	BnC (Default)	6063.7	6068.3	6084.3	0.3	9923.6	0.0	600.0
	BnC (Parity)-1	6062.4	6081.8	6081.8	0.0	32.9	16.4	23.3
	BnC (Parity)-2	6081.4	6081.8	6081.8	0.0	196.9	10.6	38.3

References

- Bachman, A., & Janiak, A. (2004). Scheduling jobs with position-dependent processing times. *Journal of the Operational Research Society*, 55(3), 257–264.
- Coffman, E. G., & Graham, R. L. (1972). Optimal scheduling for two-processor systems. *Acta Informatica*, 1(3), 200–213.
- Conforti, M., Cornuéjols, G., & Zambelli, G. (2014). *Integer programming, volume 271 of graduate texts in mathematics*. Berlin: Springer.
- Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5, 287–326.
- Hoogeveen, H., Schuurman, P., & Woeginger, G. J. (2001). Non-approximability results for scheduling problems with minsum criteria. *INFORMS Journal on Computing*, 13(2), 157–168.
- Hu, T. C. (1961). Parallel sequencing and assembly line problems. *Operations Research*, 9(6), 841–848.
- Jeroslow, R. G. (1975). On defining sets of vertices of the hypercube by linear inequalities. *Discrete Mathematics*, 11(2), 119–124.
- Johannes, B. (2005). On the complexity of scheduling unit-time jobs with or-precedence constraints. *Operations Research Letters*, 33(6), 587–596.
- Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics*, 2(1–2), 83–97.
- Kulkarni, J., & Munagala, K. (2012). Algorithms for cost-aware scheduling. In *International workshop on approximation and online algorithms* (pp. 201–214). Springer, Berlin.
- Lancia, G., & Serafini, P. (2018). *Compact extended linear programming models*. Berlin: Springer.
- Lawler, E. L. (1978). Sequencing jobs to minimize total weighted completion time subject to precedence constraints. In B. Alspach, P. Hell, & D. J. Miller (Eds.), *Annals of discrete mathematics* (Vol. 2, pp. 75–90). Amsterdam: Elsevier.
- Lenstra, J. K., & Rinnooy Kan, A. (1978). Complexity of scheduling under precedence constraints. *Operations Research*, 26(1), 22–35.
- Lenstra, J. K., & Rinnooy Kan, A. (1980). Complexity results for scheduling chains on a single machine. *European Journal of Operational Research*, 4(4), 270–275.
- Leung, J. Y.-T., & Young, G. H. (1990). Minimizing total tardiness on a single machine with precedence constraints. *ORSA Journal on Computing*, 2(4), 346–352.
- Möhring, R. H., Skutella, M., & Stork, F. (2004). Scheduling with and/or precedence constraints. *SIAM Journal on Computing*, 33(2), 393–415.
- Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer programming and combinatorial optimization*. Chichester: Wiley.
- Rudek, R. (2012). Scheduling problems with position dependent job processing times: Computational complexity results. *Annals of Operations Research*, 196(1), 491–516.
- Timkovsky, V. G. (2003). Identical parallel machines versus unit-time shops and preemptions versus chains in scheduling complexity. *European Journal of Operational Research*, 149(2), 355–376.
- Ullman, J. D. (1975). Np-complete scheduling problems. *Journal of Computer and System sciences*, 10(3), 384–393.
- Wan, G., & Qi, X. (2010). Scheduling with variable time slot costs. *Naval Research Logistics (NRL)*, 57(2), 159–171.
- Zhao, Y., Qi, X., & Li, M. (2016). On scheduling with non-increasing time slot cost to minimize total weighted completion time. *Journal of Scheduling*, 19(6), 759–767.