PACKING, PARTITIONING, AND COVERING SYMRESACKS

CHRISTOPHER HOJNY

Abstract. In this paper, we consider symmetric binary programs that contain set packing, partitioning, or covering inequalities. To handle symmetries as well as set packing, partitioning, or covering constraints simultaneously, we introduce *constrained symresacks* which are the convex hull of all binary points that are lexicographically not smaller than their image w.r.t. a coordinate permutation and which fulfill packing, partitioning, or covering constraints. We show that linear optimization problems over constrained symresacks can be solved in cubic time. Furthermore, we derive complete linear descriptions of constrained symresacks for particular classes of symmetries. These inequalities can then be used as strong symmetry handling cutting planes in a branch-and-bound procedure. Numerical experiments show that we can benefit from incorporating set packing, partitioning, or covering constraints into symmetry handling inequalities.

1. Introduction

Symmetries in binary programs typically slow down branch-and-bound procedures since symmetric solutions are computed repeatedly without providing new information to the solver. A standard approach to handle symmetries is to add cutting planes to the binary program that enforce solutions to be lexicographically maximal in their symmetry class/orbit, see, e.g., Friedman [7] or Liberti [26]. If the symmetries form a group Γ that reorders the columns of binary $(m \times n)$-matrices, Kaibel and Pfetsch [20] introduced the concept of full orbitopes to handle these symmetries. The *full orbitope* $\mathrm{O}_{m,n}(\Gamma)$ is the convex hull of all binary matrices whose columns are sorted lexicographically maximal w.r.t. reorderings by permutations in Γ. Thus, inequalities that are valid for orbitopes cut off solutions that are not lexicographically maximal in their orbit. For this reason, these inequalities can be used to handle symmetries in binary programs. Unfortunately, complete linear descriptions of full orbitopes, and thus strong symmetry handling inequalities, are unknown in general.

In applications like coloring problems, however, only those vertices of full orbitopes are of interest that have at most or exactly one 1-entry per row. Incorporating this property into orbitopes leads to so-called *packing* and *partitioning orbitopes*, see Kaibel and Pfetsch [20]. In the case of Γ being the symmetric or cyclic group, Kaibel and Pfetsch were able to describe packing and partitioning orbitopes completely. Thus, the strongest symmetry handling inequalities that incorporate the additional problem structure are known in these cases. For orbitopes with at most $k \geq 2$ ones per row or orbitopes endowed with covering constraints, however, the separation problem is NP-hard, see Loos [27]. Thus, the corresponding symmetry handling inequalities cannot be separated in polynomial time unless $\mathrm{P} = \mathrm{NP}$.

While orbitopes are only applicable if the symmetric binary program has a certain structure, a different approach for deriving symmetry handling inequalities for arbitrary binary programs was discussed in [14]. A coordinate permutation γ is a *symmetry* of a binary program if permuting the coordinates of a binary vector x by γ leads to a vector $\gamma(x)$ that is feasible for the binary program if and only if x is feasible and both x and $\gamma(x)$ have the same objective value. To handle the symmetry γ, we have introduced in [14] the *symresack* P_γ which is the convex hull of all binary vectors that are lexicographically not

Date: November 2019.

smaller than their images w.r.t. γ. Similar to orbitopes, valid inequalities for symresacks can be used to handle symmetries in binary programs. In particular, if we are given an integer programming (IP) formulation of P_γ, we can completely handle the symmetries of γ. In [14], we derived such an IP formulation of exponential size that can be separated in almost linear time, and we used this concept to derive an IP formulation of full orbitopes that can be separated in linear time. However, complete linear descriptions of P_γ are unknown in general.

Contribution and Outline. Based on the idea of packing and partitioning orbitopes, the aim of this paper is to extend the framework of symresacks for deriving general symmetry handling inequalities by taking additional constraints into account. The goal of this is to obtain a mechanism for generating strong symmetry handling cutting planes that exploit the additional structure incorporated into symresacks. To allow for a wide scope of application, we consider symresacks P_γ endowed with additional cardinality constraints, since these constraints appear in many applications. The considered cardinality constraints are of type $\sum_{i \in I} x_i \sim k$, where I is the support of a cycle of γ, k is a positive integer, and $\sim \in \{\leq, =, \geq\}$. They cover the case of upper and lower bound constraints as well as equality constraints, and we will particularly focus on the case $k = 1$. Although the condition that I is a subset of a cycle's support seems to be restrictive, it is naturally fulfilled in coloring and assignment problems as we will see in Section 6.

After formally introducing cardinality constrained symresacks (Section 2), we develop an optimization algorithm for cardinality constrained symresacks with general cardinality bound k that runs in cubic time (Section 3). This result allows us to derive a mechanism to separate inequalities handling both symmetries and cardinality constraints in polynomial time. Thus, the concept of symresacks enables us to handle symmetries with cardinality bound $k \geq 2$ efficiently, while the related approach via orbitopes is NP-hard.

Unfortunately, this result is only of theoretical interest because of two reasons. First, although the running time of the separation routine is polynomial, it might still be very large. Second, using the derived mechanism, we have no control on the kind of inequalities that we separate. For example, it is possible that the separation routine generates inequalities with huge coefficients that might cause numerical instabilities. To avoid these disadvantages, it is inevitable to develop refined methods.

For this reason, we concentrate in the remainder of this paper on the special case where $k = 1$, i.e., the cardinality constraints are so-called set packing, partitioning, and covering constraints. For the packing and partitioning case (Section 4), we develop a linear size integer programming formulation of the constrained symresack all of whose coefficients are either 0 or ± 1. That is, we are no longer relying on exponentially many inequalities as in the IP formulation of unconstrained symresacks. In particular, we prove for an exponentially large class of symmetries that this IP formulation provides already a complete linear description of packing and partitioning symresacks. Thus, the strongest symmetry handling inequalities incorporating packing and partitioning constraints are known in this case. For the covering case (Section 5), we are able to find small complete linear descriptions of the corresponding symresacks for certain symmetries. Consequently, incorporating additional structure into symresacks allows to find complete linear descriptions, while such descriptions are unknown for the unconstrained symresack.

Numerical experiments (Section 6) show that the symmetry handling inequalities based on constrained symresacks are much more effective than the symmetry handling inequalities based on unconstrained symresacks. The newly derived inequalities are also competitive with the established method orbital fixing, see Margot [29, 30] and Ostrowski [35], and even outperforming it on certain instances. In particular, since linearly many inequalities suffice to handle such symmetries, these inequalities can be added initially to the problem formulation and it is not necessary to implement a separation routine for symmetry handling inequalities. An outline on possible future research concludes the paper (Section 7).

2. Symresacks and Cardinality Constrained Symresacks

In this section, we provide basic definitions and concepts for handling symmetries in binary programs. Moreover, we give a formal definition of symresacks and rephrase some properties of symresacks from [14]. These properties will be useful in our investigation of cardinality constrained symresacks, which will be defined below.

A *symmetry* of a binary program

$$\max\left\{w^\top x :\ Ax \le b,\ x \in \{0,1\}^n\right\}, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $w \in \mathbb{R}^n$, is a permutation γ of the set $[n] \coloneqq \{1,\dots,n\}$ with the following two properties: First, $\gamma(x) \coloneqq (x_{\gamma^{-1}(1)},\dots,x_{\gamma^{-1}(n)})$ fulfills the constraints of (1) if and only if x fulfills these constraints, i.e., γ transforms feasible solutions to feasible solutions. Second, γ keeps the objective value invariant, i.e., $w^\top x = w^\top \gamma(x)$. The set of all symmetries of (1) forms a permutation group, the so-called *symmetry group* of (1), and is a subgroup of $\mathcal{S}_n$, the group containing all permutations of $[n]$. Since computing the symmetry group of a binary program is NP-hard, see Margot [31], one often refrains from computing the whole symmetry group. Instead, one typically considers subgroups Γ of the symmetry group that keep the problem formulation invariant, and only handles the symmetries contained in Γ. In the following, we always assume that Γ is a subgroup of the symmetry group of a binary program.

Because permutations $\gamma \in \Gamma$ map feasible solutions x of a binary program to feasible solutions $\gamma(x)$ preserving the objective value, it is not necessary to compute different symmetric solutions in a branch-and-bound procedure. Instead, it suffices to compute at most one representative x of its symmetry class/orbit $\{\gamma(x) :\ \gamma \in \Gamma\}$. A system of orbit representatives is called a *fundamental domain* of a binary program.

Before the idea to restrict the set of solutions to a fundamental domain came up in integer programming, it was already used in the field of constraint satisfaction problems to improve the performance of solvers, see, e.g., Aloul et al. [1] or Katsirelos et al. [22]. On the negative side, Luks and Roy [28] proved that exponentially large constraint systems may be necessary to enforce a solution to be contained in a fundamental domain. That is, no small symmetry reduction schemes may be available.

To use the idea of fundamental domains in binary programs, Friedman [7] considered the *fundamental domain inequalities (FD-inequalities)*

$$\sum_{i=1}^{n}(2^{n-\gamma(i)} - 2^{n-i})x_i \le 0, \quad \gamma \in \Gamma. \tag{2}$$

He showed that a vector $x \in \{0,1\}^n$ fulfills Inequality (2) if and only if x is not lexicographically smaller than its permutation $\gamma(x)$, denoted $x \succeq \gamma(x)$. Thus, by adding Inequality (2) for every $\gamma \in \Gamma$ to a binary program, only those solutions remain feasible that are lexicographically maximal in their orbits. The so obtained fundamental domain is of minimum size because the lexicographic order is a total order, and thus, the maximal element in an orbit is unique. Consequently, FD-inequalities remove all the symmetry w.r.t. Γ from a binary program.

Note that Friedman's approach neglects the group structure of the symmetry group Γ, because every symmetry $\gamma \in \Gamma$ is treated separately by an FD-inequality. This shows that it is not necessary to analyze the interplay of different permutations (which can be rather complex) to derive symmetry handling inequalities. From an application orientated point of view, however, the FD-inequalities used by Friedman are impractical since their large coefficients might cause numerical problems. To overcome these numerical issues, we introduced symresacks in [14].

Definition 1. Let $\gamma \in \mathcal{S}_n$. The *symresack* w.r.t. γ is the polytope

$$\mathrm{P}_\gamma \coloneqq \mathrm{conv}\left(\left\{x \in \{0,1\}^n : \sum_{i=1}^{n}(2^{n-\gamma(i)} - 2^{n-i})x_i \leq 0\right\}\right),$$

i.e., the vertices of P_γ are the binary vectors x being lexicographically not smaller than $\gamma(x)$.

To clarify the name "symresack", observe that symresacks are knapsack polytopes because they are defined by a single linear inequality and binary constraints. Moreover, symresacks P_γ can be used to handle the symmetries of a single permutation γ since valid inequalities for symresacks enforce a binary vector x to fulfill $x \succeq \gamma(x)$. Hence, the analysis of symresacks allows to derive symmetry handling inequalities. In particular, the strongest symmetry handling inequalities for a single permutation γ can be obtained by deriving a facet description of P_γ. Moreover, the inequalities of any IP formulation of P_γ have the same symmetry reducing effect as FD-inequalities. Thus, by finding an IP formulation of P_γ whose inequalities have small coefficients, we can avoid the numerical instabilities that may arise when using FD-inequalities.

To find such an IP formulation, one can exploit that P_γ is a knapsack polytope. Note that symresacks are not classical knapsack polytopes because (2) has positive and negative coefficients. However, P_γ can easily be transformed into a classical knapsack polytope by applying the map $x_i \mapsto 1 - x_i$ to all variables x_i with a negative coefficient in (2). Thus, the theory for knapsack polytopes is applicable for (transformed) symresacks. In particular, the famous IP formulation of knapsack polytopes via so-called minimal cover inequalities, see Balas and Jeroslow [4], can be used to obtain an IP formulation of P_γ all of whose left-hand side coefficients are either 0 or ± 1. Hence, we can avoid the numerical instabilities caused by FD-inequalities by separating minimal cover inequalities. While this separation problem is NP-hard for general knapsacks, see Klabjan et al. [24], we proved in [14] that separating minimal cover inequalities for symresacks is possible in $\mathcal{O}(n\alpha(n))$ time, where $\alpha\colon \mathbb{Z}_+ \to \mathbb{Z}_+$ is the inverse Ackermann function.

As a consequence, all symmetries of a binary program can be handled by replacing the FD-inequalities by the corresponding symresacks P_γ for every $\gamma \in \Gamma$ and separating minimal cover inequalities of symresacks, which is possible in $\mathcal{O}(|\Gamma| n\alpha(n))$ time.

In the following, we add cardinality restrictions to symresacks to find symmetry handling inequalities that take cardinality bounds into account. These inequalities are expected to be stronger than inequalities derived from unconstrained symresacks. For the remainder of this paper, we assume that a permutation $\gamma \in \mathcal{S}_n$ is given via its disjoint cycle decomposition $\gamma = \zeta_1 \circ \dots \circ \zeta_q$. We denote the number of cycles in the disjoint cycle decomposition of γ by $q = q(\gamma)$. The *support* of a cycle ζ_ℓ is denoted by $Z_\ell \coloneqq \{i \in [n] : \zeta_\ell(i) \neq i\}$.

Definition 2. Let $\gamma \in \mathcal{S}_n$ and let $k \in \mathbb{Z}_+^q$. The *k-packing*, *k-partitioning*, and *k-covering symresacks* w.r.t. γ are

$$\mathrm{P}_\gamma^{\leq k} \coloneqq \mathrm{conv}\left(\left\{x \in \mathrm{P}_\gamma \cap \{0,1\}^n : \sum_{i \in Z_\ell} x_i \leq k_\ell,\ \ell \in [q]\right\}\right),$$

$$\mathrm{P}_\gamma^{= k} \coloneqq \mathrm{conv}\left(\left\{x \in \mathrm{P}_\gamma \cap \{0,1\}^n : \sum_{i \in Z_\ell} x_i = k_\ell,\ \ell \in [q]\right\}\right), \text{ and}$$

$$\mathrm{P}_\gamma^{\geq k} \coloneqq \mathrm{conv}\left(\left\{x \in \mathrm{P}_\gamma \cap \{0,1\}^n : \sum_{i \in Z_\ell} x_i \geq k_\ell,\ \ell \in [q]\right\}\right),$$

respectively.

Note that, since $k \in \mathbb{Z}_+^q$, we allow each cycle to be constrained by a different cardinality bound. In particular, if $k_\ell = |Z_\ell|$ for every $\ell \in [q(\gamma)]$, the k-packing symresack $\mathrm{P}_\gamma^{\leq k}$ coincides with the ordinary symresack P_γ, and if k is the null vector, then $\mathrm{P}_\gamma^{\geq k} = \mathrm{P}_\gamma$.

The topic of the next section is the investigation of the optimization problem over k-packing, k-partitioning, and k-covering symresacks as well as their extension complexities. Our analysis is based on some ideas that were used to find an efficient optimization algorithm for unconstrained symresacks. In the following, we provide the main properties used in [14] to derive the optimization algorithm for symresacks.

Let $\gamma = \zeta_1 \circ \cdots \circ \zeta_q \in \mathcal{S}_n$. A vector $x \in \{0,1\}^n$ is contained in P_γ if and only if x is lexicographically not smaller than $\gamma(x)$. That is, either there exists an index $c \in [n]$ such that $x_c = 1$, $\gamma(x)_c = 0$, and $x_i = \gamma(x)_i$ for every $i \in [c-1]$, or $x = \gamma(x)$. In the following, we indicate $x = \gamma(x)$ by the artificial index $c = n+1$. The index c of a vertex x is called the *critical index* of x

The idea to derive an efficient optimization algorithm for symresacks in [14] is to partition the vertices into the sets $V^c = \{x \in \mathrm{P}_\gamma \cap \{0,1\}^n : x \text{ has critical index } c\}$, $c \in [n+1]$, and to derive structural properties of the vertices in V^c. These properties are illustrated in Example 3 below. The first property is that certain entries of $x \in V^c$ are fixed to 1: If the critical index of vertex x is c, then $x_c = 1$. Since $x_c = \gamma(x)_{\gamma(c)}$, we can also deduce the value of an entry in $\gamma(x)$. If $\gamma(c) < c$, then $\gamma(x)_{\gamma(c)} = x_{\gamma(c)}$ as x and $\gamma(x)$ are identical in entries in front of the critical index. Hence, further entries of x (and $\gamma(x)$) have the same value as x_c. This mechanism of applying γ to c can be iteratively used until we end up with an index that is as least as large as c, which leads to a set of indices F_c such that $x_i = 1$ for every $i \in F_c$. Since the set F_c is obtained by an iterated application of γ to c, there exists a cycle ζ with support Z of γ such that $F_c \subseteq Z$.

Analogously, one can deduce (i) from $\gamma(x)_c = 0$ a set $F_{\gamma^{-1}(c)}$ of indices with $x_i = 0$ for every $i \in F_{\gamma^{-1}(c)}$; (ii) for every $i \notin I = F_c \cup F_{\gamma^{-1}(c)}$ a set F_i such that $x_i = x_j$ for all $j \in F_i$, however, we do not know the value of x_i yet; (iii) there are no dependencies between variables from different sets F_i and F_j.

The optimization algorithm then iterates over the variables $i \in I$, computes the sets F_i, and determines whether is it beneficial to set one variable (and thus all variables) in F_i to 0 or 1. By applying this procedure for every value $c \in [n+1]$, the optimization problem over symresacks can be solved. Note that if $F_c \cap F_{\gamma^{-1}(c)} \neq \varnothing$, the values $x_c = 1$ and $\gamma(x)_c = 0$ are in conflict and V^c is empty in this case.

Example 3 (Example from Section A in [14])**.** Let $\gamma = (1,4)(2,5,7,8,3,6)$. Figure 1(a) provides in the first row the coordinates of an *original* vector $x \in \{0,1\}^8$ and in the second row the permuted entries in the *permuted* vector $\gamma(x)$. Consider the case of critical index $c = 5$, i.e., all columns in front of column 5 are of type $\binom{0}{0}$ or $\binom{1}{1}$.

Due to $c = 5$, we have $x_5 = 1$. Since we can find the value of x_5 at position 7 of $\gamma(x)$ (as $\gamma(5) = 7$), we also know $\gamma(x)_7 = 1$. Since 7 is greater than the critical index, this has no further implications on a value in x. Due to the critical index 5, we also know $\gamma(x)_5 = 0$. Since we can find the value of $\gamma(x)_5$ at position $\gamma^{-1}(5) = 2$ in x, we also know $x_2 = 0$. Since 2 is less than the critical index, this means $x_2 = \gamma(x)_2$. As $\gamma(x)_2 = x_{\gamma^{-1}(2)} = x_6$, we can also conclude $x_6 = 0$. Since 6 is greater than the critical index, this has no further implications on a value in x. Both sets of implications are shown in Figure 1(b).

For the remaining indices, we can proceed analogously and obtain the sets $F_1 = \{1,4\}$, $F_3 = \{3,8\}$, and $F_7 = \{7\}$ of entries that have to attain the same value in any vertex of P_γ with critical index 5, see Figure 1(c).

Details and correctness of this construction can be found in [14, Theorem 23]; a summary is provided in the following property.

Property 4. *The vertex set of* P_γ *can be partitioned into disjoint sets* V^c*,* $c \in [n+1]$*, with the following properties.*

(1) For every V^c*,* $c \in [n+1]$*, there exists a partition* $(F^c_j)_j$ *of* $[n]$ *such that*
- *every set* F^c_j *is a subset of a set* Z_ℓ *for some* $\ell \in [q]$*,*

1	2	3	4	5	6	7	8
4	6	8	1	2	3	5	7

(a) Original and permuted vector.

	0			1	0		
	0			0		1	

(b) Fixings induced by critical index $c = 5$.

F_1	0	F_3	F_1	1	0	F_7	F_3
F_1	0	F_3	F_1	0	F_3	1	F_7

(c) Sets of variables with the same value.

FIGURE 1. Illustration of the vertex partition V^5 of a particular symresack.

- *V^c is empty if and only if $F^c_{j(c)} = F^c_{j(\gamma^{-1}(c))}$, where $j(i)$ assigns every $i \in [n]$ the index $j(i)$ such that $i \in C^c_{j(i)}$, and*
- *computing the partition $(F^c_j)_j$ and checking $F^c_{j(c)} = F^c_{j(\gamma^{-1}(c))}$ can be done in $\mathcal{O}(n)$ time.*

(2) Let $\mathcal{F}^c \coloneqq \{F^c_{j(i)} : i \in [n]\}$. If $V^c \neq \varnothing$, every map $\phi\colon \mathcal{F}^c \to \{0,1\}$ corresponds to a vertex $x^\phi \in V^c$ via

$$x^\phi_i = \begin{cases} 1, & \text{if } i \in F^c_{j(c)},\\ 0, & \text{if } i \in F^c_{j(\gamma^{-1}(c))},\\ \phi(F^c_{j(i)}), & \text{otherwise}. \end{cases}$$

Moreover, for every vertex $x \in V^c$, there exists a map $\phi\colon \mathcal{F}^c \to \{0,1\}$ with $x = x^\phi$.

To keep notation simple, we write F^c_i instead of $F^c_{j(i)}$ in the following and identify the sets F^c_i and $F^c_{i'}$ with each other whenever $j(i) = j(i')$.

3. Optimization Complexity and Extended Formulations

In [14], we have proved that the linear optimization problem over P_γ can be solved in almost linear time for every permutation $\gamma \in \mathcal{S}_n$. Thus, symmetries of a single permutation can be handled efficiently by separating valid inequalities for P_γ due to the equivalence of optimization and separation, see Grötschel et al. [12]. To examine the complexity of handling both symmetries and additional cardinality constraints, we analyze the linear optimization problem over cardinality constrained symresacks in the following. In our analysis, we use the notation $\hat{k} \coloneqq \max_{\ell \in [q]} k_\ell$ and $\check{k} \coloneqq \min_{\ell \in [q]} k_\ell$ for a given vector $k \in \mathbb{Z}^q_+$.

Theorem 5. *Let $\gamma \in \mathcal{S}_n$ and $k \in \mathbb{Z}^q_+$. The linear optimization problem over*

- *$\mathrm{P}^{\leq k}_\gamma$ can be solved in $\mathcal{O}(n^2 \hat{k})$ time,*
- *$\mathrm{P}^{= k}_\gamma$ can be solved in $\mathcal{O}(n^2 \min\{\hat{k}, n - \check{k}\})$ time, and*
- *$\mathrm{P}^{\geq k}_\gamma$ can be solved in $\mathcal{O}(n^2 (n - \check{k}))$ time.*

Proof. Let $(V^c)_{c \in [n+1]}$ be the partition of the vertices of P_γ mentioned in Property 4. Since the constrained symresacks $\mathrm{P}^{\sim k}_\gamma$, $\sim \in \{\leq, =, \geq\}$, are subpolytopes of P_γ and all of their vertices are binary, the family $\mathrm{P}^{c, \sim k}_\gamma \coloneqq V^c \cap \mathrm{P}^{\sim k}_\gamma$, $c \in [n+1]$, is a partition of the vertices of $\mathrm{P}^{\sim k}_\gamma$. Hence, a maximizer of a linear objective $w^\top x$, where $w \in \mathbb{R}^n$, over $\mathrm{P}^{\sim k}_\gamma$ can be found by maximizing $w^\top x$ over the sets in this partition and taking a weight maximal solution. Consequently, the theorem holds if we can show that the optimization problem over $\mathrm{P}^{c, \leq k}_\gamma$, $\mathrm{P}^{c, = k}_\gamma$, and $\mathrm{P}^{c, \geq k}_\gamma$ can be solved in $\mathcal{O}(n\hat{k})$, $\mathcal{O}(n \min\{\hat{k}, n - \check{k}\})$, and $\mathcal{O}(n(n - \check{k}))$ time, respectively.

Let $c \in [n+1]$ and let $(F^c_j)_{j \in [n]}$ be the partition of V^c described in Property 4. Due to Property 4, $(F^c_j)_j$ can be constructed and emptiness of V^c (and thus $\mathrm{P}^{c, \sim k}_\gamma$) can be decided in linear time. As a consequence, we can assume in the following that $V^c \neq \varnothing$. To prove the complexity bounds for the sets $\mathrm{P}^{c, \sim k}_\gamma$, we use the second part of Property 4, which states that variables whose indices are contained in F^c_c are fixed to 1 and variables with index in $F^c_{\gamma^{-1}(c)}$ are fixed to 0. Furthermore, variables whose indices are contained in the same set F^c_j have to obtain the same value from $\{0,1\}$, but there are no implications between variables in different sets. Moreover, Property 4 implies that each set F^c_j is a subset of a cycle Z_ℓ of γ for some $\ell \in [q(\gamma)]$.

For $\ell \in [q]$, let $\mathcal{F}^c_\ell$ consist of the sets F^c_j that are contained in Z_ℓ. We introduce a binary variable y_F for every $F \in \mathcal{F}^c_\ell$ to model that all variables x_i with $i \in F$ obtain the same binary value, i.e., $x_i = y_F$. Since there does not exist a dependence between variables of different sets F^c_j and each such set is contained in exactly one cycle, the solutions on the different cycles of γ can be computed independently. Thus, the maximization problem over $\mathrm{P}^{c,\sim k}_\gamma$ decomposes into the maximization problems

$$\begin{aligned} &\max_{y \in \{0,1\}^{\mathcal{F}^c_\ell}} \Big\{ \sum_{F \in \mathcal{F}^c_\ell} w(F) y_F \,:\, \sum_{F \in \mathcal{F}^c_\ell} |F| y_F \sim k_\ell,\ y_{F^c_c} = 1,\ y_{F^c_{\gamma^{-1}(c)}} = 0, \Big\}, && \text{if } F^c_c \in \mathcal{F}^c_\ell, \\ &\max_{y \in \{0,1\}^{\mathcal{F}^c_\ell}} \Big\{ \sum_{F \in \mathcal{F}^c_\ell} w(F) y_F \,:\, \sum_{F \in \mathcal{F}^c_\ell} |F| y_F \sim k_\ell \Big\}, && \text{otherwise,} \end{aligned} \tag{3}$$

where $\ell \in [q]$ and $w(F) \coloneqq \sum_{i \in F} w_i$.

For $\mathrm{P}^{c,\leq k}_\gamma$, each of the above subproblems is a knapsack problem with knapsack inequality $\sum_{F \in \mathcal{F}^c_\ell} |F| y_F \leq k_\ell$. Since $|\mathcal{F}^c_\ell| \leq |Z_\ell|$, each subproblem contains at most $|Z_\ell|$ variables. Hence, it can be solved via dynamic programming techniques in $\mathcal{O}(|Z_\ell| k_\ell)$ time, see Kellerer et al. [23]. Consequently, the knapsack subproblems on the different cycles can be solved in $\mathcal{O}\big(\sum_{\ell \in [q]} |Z_\ell| k_\ell\big) \subseteq \mathcal{O}\big(n\hat{k}\big)$ time, because the cycle supports Z_ℓ form a partition of $[n]$.

For $\mathrm{P}^{c,\geq k}_\gamma$, let $a^\top y \coloneqq \sum_{F \in \mathcal{F}^c_\ell} |F| y_F \geq k_\ell$ be the knapsack constraint of a knapsack subproblem (3). By negating the variables, the knapsack problem can be transformed into a knapsack with $\leq$-constraint:

$$\begin{aligned} &\max_{y \in \{0,1\}^{\mathcal{F}^c_\ell}} \Big\{ \sum\nolimits_{F \in \mathcal{F}^c_\ell} w(F) y_F \,:\, a^\top y \geq k_\ell \Big\} \\ = &\max_{\tilde{y} \in \{0,1\}^{\mathcal{F}^c_\ell}} \Big\{ \sum\nolimits_{F \in \mathcal{F}^c_\ell} w(F) - \sum\nolimits_{F \in \mathcal{F}^c_\ell} w(F) \tilde{y}_F \,:\, a^\top (\mathbb{1} - \tilde{y}) \geq k_\ell \Big\} \\ = &\max_{\tilde{y} \in \{0,1\}^{\mathcal{F}^c_\ell}} \Big\{ \sum\nolimits_{F \in \mathcal{F}^c_\ell} w(F) - \sum\nolimits_{F \in \mathcal{F}^c_\ell} w(F) \tilde{y}_C \,:\, a^\top \tilde{y} \leq a^\top \mathbb{1} - k_\ell \Big\}. \end{aligned}$$

Since $a^\top \mathbb{1} = |Z_\ell|$, the subproblem on cycle ζ_ℓ can be solved via dynamic programming in time $\mathcal{O}\big(|Z_\ell|(|Z_\ell| - k_\ell)\big)$. For this reason, all subproblems can be solved in

$$\mathcal{O}\Big(\sum\nolimits_{\ell \in [q]} |Z_\ell|(|Z_\ell| - k_\ell)\Big) = \mathcal{O}\Big(\sum\nolimits_{\ell \in [q]} |Z_\ell|^2 - \sum\nolimits_{\ell \in [q]} |Z_\ell| k_\ell\Big) \subseteq \mathcal{O}\Big(n^2 - n\check{k}\Big)$$

time.

Finally, since an equality constrained knapsack problem can be solved by the techniques for either of the above knapsacks, the optimization problem over $\mathrm{P}^{c,=k}_\gamma$ can be solved $\mathcal{O}\big(n \min\{\hat{k}, n - \check{k}\}\big)$ time, which concludes the proof. □

Due to the equivalence of optimization and separation, Theorem 5 implies that valid inequalities for constrained symresacks, and thus symmetry handling inequalities which incorporate cardinality constraints, can be separated in polynomial time. Moreover, the optimization algorithm for constrained symresacks allows to derive a compact extended formulation for these polytopes. An *extended formulation* of a polyhedron $P \subseteq \mathbb{R}^n$ is a polyhedron $Q \subseteq \mathbb{R}^d$ endowed with an affine map $\pi \colon \mathbb{R}^d \to \mathbb{R}^n$ such that $P = \pi(Q)$. The size of an extended formulation is the number of inequalities needed in an outer description of Q; the *extension complexity* of a polyhedron P is the minimum size of an extended formulation of P.

Theorem 6. *Let $\gamma \in \mathcal{S}_n$ and $k \in \mathbb{Z}_+^q$. The extension complexity of*

- $\mathrm{P}_\gamma^{\leq k}$ *is in* $\mathcal{O}\left(n^2 \hat{k}\right)$,
- $\mathrm{P}_\gamma^{= k}$ *is in* $\mathcal{O}\left(n^2 \min\{\hat{k}, n - \check{k}\}\right)$, *and*
- $\mathrm{P}_\gamma^{\geq k}$ *is in* $\mathcal{O}\left(n^2 (n - \check{k})\right)$.

Proof. Given a collection $(P^c)_c$ of polytopes, the concept of disjunctive programming allows to define an extended formulation of $P := \mathrm{conv}(\bigcup_c P^c)$ by combining complete linear descriptions, see Balas [3], or extended formulations of P^c, see Kaibel and Loos [19]. In particular, if the linear description/extended formulation of P^c contains s^c inequalities, the derived extended formulation of P consists of $\mathcal{O}(\sum_c s^c)$ inequalities. Thus, since

$$\mathrm{P}_\gamma^{\leq k} = \mathrm{conv}\left(\bigcup_{c=1}^{n+1} \mathrm{conv}(\mathrm{P}_\gamma^{c,\leq k})\right),$$

we can use disjunctive programming to find an extended formulation of $\mathrm{P}_\gamma^{\leq k}$ provided that we know extended formulations of $\mathrm{conv}(\mathrm{P}_\gamma^{c,\leq k})$. To find these extended formulations, we use a classical result of Martin et al. [32].

Given a combinatorial optimization problem that can be solved by a dynamic programming algorithm, Martin et al. [32] constructed a linear programming (LP) formulation of the combinatorial problem, provided that the dynamic programming routine can be modeled as a flow problem in a suitable hypergraph. This LP model can be interpreted as an (extended) formulation of the convex hull of the feasible region of the combinatorial problem in a suitable space. Since the optimization problem over $\mathrm{P}_\gamma^{c,\leq k}$ can be solved by a dynamic programming approach, see the proof of Theorem 5, it suffices to construct the corresponding hypergraph to obtain an extended formulation of $\mathrm{conv}(\mathrm{P}_\gamma^{c,\leq k})$.

Recall that the optimization problem over $\mathrm{P}_\gamma^{c,\leq k}$ decomposes into $q(\gamma)$ many knapsack problems (3). Due to Conforti et al. [6], there exists a hypergraph G_ℓ corresponding to the feasible region of the ℓ-th knapsack problem $a^\top x \leq k_\ell$, $x \in \{0,1\}^n$, in the Decomposition (3) whose size is in $\mathcal{O}\big(|Z_\ell|\hat{k}\big)$ and which meets the requirements of Martin et al. Because the Subproblems (3) can be solved independently, we can connect the hypergraphs G_ℓ, $\ell \in [q]$, in series to model the optimization problem over $\mathrm{P}_\gamma^{c,\leq k}$ (and thus over $\mathrm{conv}(\mathrm{P}_\gamma^{c,\leq k})$) as a flow problem in a hypergraph G. Because G consists of $\mathcal{O}\big(\sum_{\ell \in [q]} |Z_\ell|\hat{k}\big) = \mathcal{O}\big(n\hat{k}\big) =: \mathcal{O}\left(s^c\right)$ nodes and arcs, the LP model of Martin et al. is of size $\mathcal{O}\left(s^c\right)$ as well, see Conforti et al. [6, Theorem 7.1]. Thus, there exists an extended formulation of $\mathrm{P}_\gamma^{\leq k}$ of size $\mathcal{O}(\sum_{c=1}^{n+1} n\hat{k}) = \mathcal{O}(n^2\hat{k})$ by the initial arguments in this proof.

To show the assertion for $\mathrm{P}_\gamma^{=k}$ and $\mathrm{P}_\gamma^{\geq k}$, we exploit that the dynamic programming approach for the Subproblems (3) can be solved by dynamic programming algorithms for $\leq$-knapsack constraints whose right-hand sides are bounded from above by $\min\{\hat{k}, n - \check{k}\}$ and $n - \check{k}$, respectively, see the proof of Theorem 5. Hence, since the size of the hypergraph constructed above only depends on the size of the cycles ζ_ℓ of γ and the right-hand side of the $\leq$-knapsack constraints, the above argumentation can directly be used to obtain extended formulations of the proposed size for $\mathrm{P}_\gamma^{=k}$ and $\mathrm{P}_\gamma^{\geq k}$, which concludes the proof. ☐

Remark 7. For the all-ones vector $k = \mathbb{1}$, Theorems 5 and 6 can be strengthened, because one can show that the linear optimization over $\mathrm{P}_\gamma^{\geq k}$ is solvable in $\mathcal{O}\left(n^2\right)$ time and that k-covering symresacks admit an extended formulation of $\mathcal{O}\left(n^2\right)$ size. To get the optimization result, we just have to iterate twice over all sets $F \in \mathcal{F}_\ell^c$ for all $c \in [n+1]$ and $\ell \in [q]$. During the first iteration, we compute the weights $w(F)$ and set $y_F = 1$ if $w(F) \geq 0$. If, for each cycle ζ_ℓ, there exists $F \in \mathcal{F}_\ell^c$ with $y_F = 1$, all cardinality constraints are satisfied.

Otherwise, we have to select for cycles ζ_ℓ in which all y-variables are 0, a set $F \in \mathcal{F}^c_\ell$ with maximum weight $w(F)$. Assigning the corresponding variable value 1 obviously produces a feasible and optimal solution. Since this procedure runs in $\mathcal{O}(n)$ time for fixed c, the running time for all values of $c \in [n+1]$ is quadratic.

To get the result on the extension complexity, let $\mathcal{F}^c = \bigcup_{\ell=1}^{q} \mathcal{F}^c_\ell$. Then, $\operatorname{conv}(\mathrm{P}^{c,\geq k}_\gamma)$ in y-variables is completely described by

$$\{y \in [0,1]^{\mathcal{F}^c} \,:\, y_{F^c_c} = 1,\ y_{F^c_{\gamma^{-1}(c)}} = 0,\ \sum_{F \in \mathcal{F}^c_\ell} y_F \geq 1,\ \ell \in [q]\}.$$

Note that this system has size $\mathcal{O}(n)$. Since the covering constraints do not overlap as the sets in $\mathcal{F}^c_\ell$ form a partition of $[n]$, this system is totally unimodular, and thus, defines an integral polytope. Hence, an extended formulation of $\mathrm{P}^{\geq \mathbb{1}}_\gamma$ can be obtained by combining the above extended formulation in y-variables for every $c \in [n+1]$ with $\mathrm{P}^{c,\geq \mathbb{1}}_\gamma \neq \varnothing$ by disjunctive programming, which has size $\mathcal{O}(n^2)$.

Remark 8. Theorems 5 and 6 can easily be generalized to the case where only some cycles of a permutation are constrained by packing, partitioning, or covering constraints or some cycles are constrained by packing constraints whereas other cycles are constrained by partitioning and covering constraints. In this case, the optimization and extension complexity of constrained symresacks is in $\mathcal{O}\left(n^3\right)$.

Since constrained symresacks admit a compact extended formulation, we are theoretically able to derive complete linear descriptions of $\mathrm{P}^{\leq k}_\gamma$, $\mathrm{P}^{=k}_\gamma$, and $\mathrm{P}^{\geq k}_\gamma$ by computing the projection of the corresponding extended formulations. However, computing this projection is quite complicated. Complete linear descriptions are hence not available in general. To be able to find strong symmetry handling inequalities that incorporate additional problem information, we thus have to analyze the corresponding constrained symresacks in more detail. In the following two sections, we therefore concentrate on the special case of cardinality constrained symresacks with $k = \mathbb{1}$, i.e., the cardinality constraints are set packing, partitioning, or covering constraints. For the sake of brevity, these cardinality constrained symresacks are called packing, partitioning, and covering symresacks instead of $\mathbb{1}$-packing, $\mathbb{1}$-partitioning, and $\mathbb{1}$-covering symresacks. The corresponding polytopes are denoted by $\mathrm{P}^{\leq}_\gamma$, $\mathrm{P}^{=}_\gamma$, and $\mathrm{P}^{\geq}_\gamma$, respectively. Moreover, observe that if γ contains a fixed point i, then γ has a cycle of length 1. Thus, the corresponding cardinality constraint is either redundant or implies that x_i is fixed to a binary value. Since both cases do not affect whether x is lexicographically greater or equal than $\gamma(x)$, we assume that γ does not contain a fixed point to simplify notation in the following two sections. Such permutations are called derangements in the literature.

4. Packing and Partitioning Symresacks

To be able to find strong symmetry handling inequalities that incorporate set packing or partitioning constraints on the cycles of a permutation $\gamma \in \mathcal{S}_n$, we investigate packing and partitioning symresacks in this section. First, we derive a linear size integer programming formulation of $\mathrm{P}^{\leq}_\gamma$ and $\mathrm{P}^{=}_\gamma$ with ternary coefficients. Afterwards, we prove that this IP formulation already defines a complete linear description for packing and partitioning symresacks of particular permutations.

For the special case of $\gamma = (1,2)(3,4)\dots(2n-1,2n)$ for some positive integer n, our inequalities reduce to the ones discussed by Ostrowski et al. [36]. Our contribution, however, is the generalization of this concept to arbitrary permutations, and to provide evidence that these generalized inequalities are the strongest possible inequalities for a certain class of permutations.

Before we start our analysis, we present an example that illustrates the application of packing and partitioning symresacks.

Example 9. Let $G = (V, E)$ be an undirected graph and let k be a positive integer. A *k-coloring* of G is an assignment of k colors to the nodes of G such that adjacent nodes are colored differently. The *maximum k-colorable subgraph problem* (MkSC) is to find a node maximum induced subgraph of G that is k-colorable. A natural IP formulation of this problem is

$$\max \sum_{v \in V} \sum_{j=1}^{k} x_{vj}$$

$$\sum_{j=1}^{k} x_{vj} \leq 1, \qquad v \in V, \tag{4a}$$

$$x_{uj} + x_{vj} \leq 1, \qquad \{u, v\} \in E,\ j \in [k], \tag{4b}$$

$$x \in \{0,1\}^{V \times [k]},$$

where $x_{vj} = 1$ if and only if node v is colored with color j.

MkCS contains two kinds of symmetries. First, we can associate with every relabeling $\gamma \in \mathcal{S}_k$ of the colors the permutation $\gamma' \in \mathcal{S}_{V \times [k]}$ that maps entry (v, j) to $(v, \gamma(j))$. Thus, each cycle of γ' is a subset of a row index set of x. Consequently, Inequality (4a) implies that there is at most one 1-entry in each cycle of γ'.

Second, each automorphism $\gamma \colon V \to V$ of G gives rise to a permutation γ' of the rows of x. Thus, each cycle of γ' is a subset of a column of x. In particular, if γ affects only nodes in a clique of G, every color can be assigned to at most one node of the affected nodes due to (4b). Thus, each non-trivial cycle of γ' contains at most one 1-entry. As a consequence, both color symmetries and graph automorphisms of cliques can be handled by packing symresacks in the above IP formulation.

Similar to the ordinary symresack P_γ, an IP formulation of constrained symresacks $\mathrm{P}_\gamma^{\leq}$ and $\mathrm{P}_\gamma^{=}$ is given by box constraints, the FD-inequality (2), and packing/partitioning constraints. To avoid the exponential coefficients of the FD-inequality in this formulation, we can of course use the IP formulation of P_γ via minimal cover inequalities and add the packing/partitioning constraints to ensure that we obtain an IP formulation for the constrained symresack. However, both the approach via the FD-inequality and via minimal cover inequalities do not combine symmetry and packing/partitioning information. Thus, it is likely that these IP formulations are weaker than an IP formulation that incorporates the structural information of packing/partitioning constraints into symmetry handling inequalities. For this reason, we analyze the vertices of $\mathrm{P}_\gamma^{\leq}$ and $\mathrm{P}_\gamma^{=}$ to be able to derive a tighter IP formulation. To this end, it will be convenient to denote the set of all *ascent points* of γ by $\mathcal{A} = \mathcal{A}^\gamma := \{i \in [n] : \gamma(i) > i\}$ and the set of all *descent points* of γ by $\mathcal{D} = \mathcal{D}^\gamma := \{i \in [n] : \gamma(i) < i\}$. Since we assume that γ does not contain a fixed point, we have $\mathcal{A} \cup \mathcal{D} = [n]$.

Example 10. Consider the permutation $\gamma = (1, 4)(2, 5, 7, 8, 3, 6)$. Suppose we are given $x \in \{0,1\}^8$ that satisfies packing constraints on both cycles of γ. If $x_8 = 1$, then $\gamma(x)_3 = x_{\gamma^{-1}(3)} = x_8 = 1$. Consequently, if no 1-entry on the first cycle of γ exists, $\gamma(x) = (0, 0, 1, 0, 0, 0, 0, 0) \succ (0, 0, 0, 0, 0, 0, 0, 1) = x$ as the unique 1-entry of x is at position 8 and the unique 1-entry of $\gamma(x)$ at position 3.

To ensure that $x \succeq \gamma(x)$, we have to assign a 1-entry to the first cycle. If this entry is x_1, then $x = (1, 0, 0, 0, 0, 0, 0, 1)^\top$ and $\gamma(x) = (0, 0, 1, 1, 0, 0, 0, 0)^\top$, which guarantees $x \succ \gamma(x)$. The reason for this is that we assign a 1-entry to an ascent point of γ that comes in front of 3, the image of the descent point 8. That is, while assigning a 1-entry to a descent point leads to an "earlier" 1-entry in $\gamma(x)$, which might lead to a lexicographically greater point, assigning a 1-entry to an ascent point in front of the descent point's image ensures that x is lexicographically at least as large as $\gamma(x)$.

The following lemma formalizes this idea and completely characterizes the vertices of $\mathrm{P}^{\leq}_{\gamma}$ and $\mathrm{P}^{=}_{\gamma}$, which will be useful in deriving a complete linear description. Note, however, that this characterization is only valid for the packing and partitioning case, i.e., $k = 1$, and cannot be used for arbitrary cardinality bounds.

Lemma 11. *Let $\gamma \in \mathcal{S}_n$ and let $\bar{x} \in \{0,1\}^n$ fulfill packing and partitioning constraints, respectively, on the cycles of γ. Then $\bar{x}$ is contained in $\mathrm{P}^{\leq}_{\gamma}$ and $\mathrm{P}^{=}_{\gamma}$, respectively, if and only if*

- *$\bar{x}_j = 0$ for each $j \in \mathcal{D}$ or*
- *for each $j \in \mathcal{D}$ with $\bar{x}_j = 1$ there exists $i < \gamma(j)$ such that $\bar{x}_i = 1$ and $i \in \mathcal{A}$.*

Proof. By definition, a vector x is lexicographically not smaller than a vector y, denoted by $x \succeq y$, if and only if either both vectors are equal or $x_i > y_i$ for the first position i in which both differ. On the one hand, if $\bar{x}_j = 0$ for each $j \in \mathcal{D}$, then $\bar{x} \succeq \gamma(\bar{x})$ since each non-zero entry of $\bar{x}$ is permuted to an entry with a larger index. Hence, $\bar{x}$ is contained in the packing or partitioning symresack.

On the other hand, assume that there exists $j \in \mathcal{D}$ with $\bar{x}_j = 1$. Let $j^\star \in \mathcal{D}$ with $\bar{x}_{j^\star} = 1$ be the index for which $\gamma(j^\star)$ is minimal. If $\bar{x}_i = 0$ for every $i \in [\gamma(j^\star) - 1] \cap \mathcal{A}$, then the first non-zero entry of $\gamma(\bar{x})$ is $\gamma(j^\star)$. Moreover, the first non-zero entry of $\bar{x}$ is greater than $\gamma(j^\star)$, since $\gamma(j^\star)$ is minimal and $j^\star$ is the only 1-entry of $\bar{x}$ on the cycle containing $j^\star$ due to the packing/partitioning constraints. Thus, $\gamma(\bar{x}) \succ \bar{x}$ and $\bar{x}$ is not contained in the constrained symresack.

Finally, if there exists $i \in [\gamma(j^\star) - 1]$ such that $\bar{x}_i = 1$, then i is an ascent point by the choice of $\gamma(j^\star)$. Let $i^\star$ be the minimal index with this property. Then the first non-zero entry of $\bar{x}$ is $i^\star$, while the first non-zero entry of $\gamma(\bar{x})$ is greater than $i^\star$ by the choice of $i^\star$ and $j^\star$ as well as the packing/partitioning constraints on the cycles. Hence, $\bar{x} \succ \gamma(\bar{x})$ follows, which proves that $\bar{x}$ is contained in the packing/partitioning symresack. □

The characterization of the vertices of $\mathrm{P}^{\leq}_{\gamma}$ and $\mathrm{P}^{=}_{\gamma}$ of Lemma 11 can easily be enforced by the inequalities

$$-\sum_{\substack{i<\gamma(j):\\ i\in\mathcal{A}^{\gamma}}} x_i + x_j \leq 0, \quad j \in \mathcal{D}^{\gamma}. \tag{5}$$

Furthermore, since all entries of a vertex x of $\mathrm{P}^{\leq}_{\gamma}$ and $\mathrm{P}^{=}_{\gamma}$ are non-negative, the packing and partitioning constraints imply that each entry of x is at most 1. Thus, it is not necessary to define the upper bound constraints $x_i \leq 1$, $i \in [n]$, and we immediately get the IP formulation of $\mathrm{P}^{\leq}_{\gamma}$ and $\mathrm{P}^{=}_{\gamma}$ stated in the following proposition.

Proposition 12. *For every $\gamma \in \mathcal{S}_n$, an IP formulation of $\mathrm{P}^{\leq}_{\gamma}$ is given by packing constraints, non-negativity constraints, and (5). Moreover, by substituting packing constraints by partitioning constraint, we obtain an IP formulation for $\mathrm{P}^{=}_{\gamma}$.*

Proposition 12 shows that symmetries related to packing and partitioning symresacks can be handled by linearly many inequalities with ternary coefficients. In contrast to ordinary symresacks, we are thus not relying on a separation routine for symmetry handling inequalities, because the Inequalities (5) can be added initially to the problem formulation without increasing the size of the problem formulation too much. In particular, we avoid the exponential coefficients of FD-inequalities.

To evaluate the strength of this IP formulation, we analyze whether it is already a complete linear description of packing and partitioning symresacks in the following. In fact, we will see that the IP formulation completely describes $\mathrm{P}^{\leq}_{\gamma}$ and $\mathrm{P}^{=}_{\gamma}$ if γ is a monotone permutation. A permutation γ is called *monotone* if every cycle of γ contains exactly one descent point, which is then the maximum element in the cycle.

Example 13. Assume that the entries of an $(m \times n)$-matrix X are ordered as

$$(1,1) < (1,2) < \cdots < (1,n) < (2,1) < \cdots < (m,n-1) < (m,n).$$

Let $\gamma \in \mathcal{S}_n$ and let $\gamma' \in \mathcal{S}_{[m]\times[n]}$ be the permutation that reorders the columns of X according to γ, cf. the color symmetries in Example 9. Then, a column reordering γ' is monotone if and only if the underlying permutation γ is monotone.

To analyze the IP formulations of $\mathrm{P}^{\leq}_{\gamma}$ and $\mathrm{P}^{=}_{\gamma}$ for monotone permutations, we define for every $\ell \in [q]$ the values $\check{z}_\ell := \min\{i \in Z_\ell\}$ and $\hat{z}_\ell := \max\{i \in Z_\ell\}$. Moreover, define $Z^{\star}_\ell := Z_\ell \setminus \{\hat{z}_\ell\}$ and $\hat{Z} := \{\hat{z}_\ell : \ell \in [q]\}$. Using this notation, Inequalities (5) simplify for monotone permutations to

$$-\sum_{i \in [\check{z}_\ell - 1] \setminus \hat{Z}} x_i + x_{\hat{z}_\ell} \leq 0, \quad \ell \in [q(\gamma)]. \tag{6}$$

For partitioning symresacks, we will see below that the following inequalities already define a complete linear description:

$$-\sum_{i=1}^{\check{z}_\ell - 1} x_i + x_{\hat{z}_\ell} \leq 0, \qquad \ell \in [q(\gamma)], \tag{7a}$$

$$\sum_{i \in Z_\ell} x_i = 1, \qquad \ell \in [q(\gamma)], \tag{7b}$$

$$-x_i \leq 0, \qquad i \in [n]. \tag{7c}$$

Observe that (7a) is slightly weaker than (6). However, the weaker inequalities suffice to define a complete linear description, since if there exists $\ell \in [q]$ such that $[\check{z}_\ell - 1] \cap \hat{Z} \neq \varnothing$, the corresponding Inequality (7a) is redundant due to the remaining inequalities (7a) and partitioning constraints.

In the following, we refer to (7a) as *ordering constraints*, to (7b) as *partitioning constraints*, and to (7c) as *non-negativity constraints*. Example 14 illustrates the structure of ordering constraints, which might be helpful in the proof of Theorem 15.

Example 14. Consider the monotone permutation

$$\gamma = (1,3,6,10)(2,7,8,11)(4,9)(5,12).$$

The ordering constraints of its four cycles are

$$x_{10} \leq 0, \quad -x_1 + x_{11} \leq 0, \quad -x_1 - x_2 - x_3 + x_9 \leq 0, \text{ and } \quad -x_1 - x_2 - x_3 - x_4 + x_{12} \leq 0.$$

In particular, if N_ℓ denotes the indices of variables that have a negative sign in the ordering constraint of cycle ℓ, then $N_\ell \subseteq N_{\ell'}$ if and only if $\check{z}_\ell \leq \check{z}_{\ell'}$.

Theorem 15. *Let $\gamma \in \mathcal{S}_n$ be a monotone permutation without fixed points. Then $\mathrm{P}^{=}_{\gamma}$ is completely described by (7).*

Proof. The result can be shown by analyzing structural properties of bases of System 7. Using these properties, one can show that no basis can define a fractional vertex, and thus, (7) is a complete linear description of $\mathrm{P}^{=}_{\gamma}$. A full proof is provided in A. □

Remark 16. In general, Theorem 15 is false if γ is not monotone. For example, consider the permutation $\gamma = (1,7,5,3)(2,8,6,4)$. Experiments with the software tool POLYMAKE [2] show that the polytope defined by (5), (7b), and (7c) has fractional vertices. Thus, the three families of inequalities are not a complete linear description of $\mathrm{P}^{=}_{\gamma}$.

Of course, every permutation γ can be transformed into a monotone one by relabeling the variables it acts on, which allows us to find a complete linear description for P_γ. However, this is not with loss of generality because if we want to use constrained symresacks in practice, we typically deal with more than one permutation. Rearranging variables then allows to obtain one monotone permutation, but not to transform every permutation simultaneously into a monotone one in general.

By a projection argument, we are also able to prove that the IP formulation of packing symresacks provided in Proposition 12 already is a complete linear description of $\mathrm{P}^{\leq}_{\gamma}$ if γ is monotone.

Proposition 17. *Let $\gamma \in \mathcal{S}_n$ be a monotone permutation with q cycles in its disjoint cycle decomposition. Then there exists a monotone permutation $\tilde{\gamma} \in \mathcal{S}_{n+q}$ such that $\mathrm{P}^{\leq}_{\gamma}$ is a linear projection of $\mathrm{P}^{=}_{\tilde{\gamma}}$.*

Proof. An explicit construction of $\tilde{\gamma}$ can be found in A. □

Theorem 18. *Let $\gamma \in \mathcal{S}_n$ be a monotone permutation without fixed points. Then $\mathrm{P}^{\leq}_{\gamma}$ is completely described by (6), (7c), and the packing constraints $\sum_{i \in Z_\ell} x_i \leq 1$ for all $\ell \in [q]$.*

Proof. The idea is to apply Fourier-Motzkin elimination to System (7) for the permutation $\tilde{\gamma}$ from Proposition 17. This leads to the proposed description of $\mathrm{P}^{\leq}_{\gamma}$, see A for details. □

Furthermore, we are able to completely characterize which of the inequalities of the complete linear description define facets of $\mathrm{P}^{\leq}_{\gamma}$.

Theorem 19. *Let $\gamma \in \mathcal{S}_n$ be a monotone permutation without fixed points and assume that $\check{z}_1 < \check{z}_2 < \cdots < \check{z}_q$. Then a facet of $\mathrm{P}^{\leq}_{\gamma}$ is defined by*

- *all packing constraints $\sum_{i \in Z_\ell} x_i \leq 1$, $\ell \in [q]$,*
- *(6) if and only if $\ell \neq 1$, and*
- *(7c) if and only if $i \in \{2, \dots, n\} \setminus \{\hat{z}_1\}$ or $i = 1$ and $\{2, \dots, \check{z}_2 - 1\} \setminus \{\hat{z}_1\} \neq \varnothing$.*

Proof. See A for details. □

Due to Theorems 15 and 18, we are able to handle symmetries related to packing and partitioning symresacks of monotone permutations not only via an IP formulation but even by a complete linear description of $\mathrm{P}^{\leq}_{\gamma}$ and $\mathrm{P}^{=}_{\gamma}$. Hence, the strongest symmetry handling inequalities that handle both packing/partitioning constraints and symmetry information are known for such permutations. As a consequence, the symmetry handling effect of ordering constraints should, at least in theory, be more effective than the effect of minimal cover inequalities for symresacks or the FD-inequality (2). In particular, using ordering constraints is numerically much more stable than the latter approach since the coefficients of ordering constraints are ternary, while coefficients in FD-inequalities grow exponentially. Moreover, we can avoid a separation routine for the exponentially many minimal cover inequalities for ordinary symresacks due to the linear size (IP) formulation of packing and partitioning symresacks.

5. Covering Symresacks

The aim of this section is to derive symmetry handling inequalities that incorporate set covering constraints. To this end, we investigate covering symresacks. Similar to Section 4, we first present an example that shows the applicability of these polytopes. Afterwards, we derive a complete linear description of $\mathrm{P}^{\geq}_{\gamma}$ for particular permutations.

Example 20. Let $\mathcal{C}$ be a collection of subsets of $[n]$ such that $\bigcup_{C \in \mathcal{C}} C = [n]$. The *set covering problem* is to find a smallest subcollection $\mathcal{C}'$ of $\mathcal{C}$ such that $\bigcup_{C \in \mathcal{C}'} C = [n]$. A classical IP formulation of this problem is

$$\min \Big\{ \sum_{C \in \mathcal{C}} x_C \,:\, \sum_{C \in \mathcal{C}_i} x_C \geq 1,\ i \in [n],\ x \in \{0,1\}^{\mathcal{C}} \Big\},$$

where $\mathcal{C}_i \coloneqq \{C \in \mathcal{C} \,:\, i \in C\}$, see Vazirani [38]. If there exists a symmetry $\gamma \in \mathcal{S}_{\mathcal{C}}$ of the IP formulation such that each support Z of a cycle of γ is a superset of a set $\mathcal{C}_i$, then

$$\sum_{C \in Z} x_C \geq \sum_{C \in \mathcal{C}_i} x_C \geq 1.$$

Hence, $\sum_{C \in Z} x_S \geq 1$ is a valid inequality for the IP formulation, and as a consequence, the symmetries related to γ can be handled by $\mathrm{P}^{\geq}_{\gamma}$.

Computer experiments with POLYMAKE [2] indicate that the facial structure of $\mathrm{P}^{\geq}_{\gamma}$ is rather complicated, and in particular, more complex than the facial structure for packing and partitioning symresacks even if the underlying permutation is monotone. In the following, we concentrate for this reason on monotone permutations that are also ordered. A permutation γ is called *ordered* if $\check{z}_{\ell+1} = \hat{z}_\ell + 1$ for all $\ell \in [q(\gamma) - 1]$, i.e., the supports of the permutation's cycles are consecutive sets. The basis of our investigation is provided by the following lemma.

Lemma 21. *Let $\bar{x} \in \{0,1\}^n$ and let $\gamma \in \mathcal{S}_n$ be a monotone and ordered permutation. Then $\bar{x} \succeq \gamma(\bar{x})$ if and only if*

- *either $\bar{x}$ is constant along each cycle of γ, i.e., $\bar{x}_i = \bar{x}_j$ for all $i, j \in Z_\ell$, $\ell \in [q]$, or*
- *if $\ell' \in [q]$ is the smallest index of a non-constant cycle of γ w.r.t. $\bar{x}$, then $\bar{x}_{\hat{z}_{\ell'}} = 0$.*

Proof. Since $\bar{x} = \gamma(\bar{x})$ clearly holds if and only if $\bar{x}$ is constant along each cycle, we can assume w.l.o.g. that there exists a non-constant cycle of γ w.r.t. $\bar{x}$. Let $\ell' \in [q]$ be the index of the first non-constant cycle, and recall that $\bar{x} \succ \gamma(\bar{x})$ if and only if $\bar{x}_{i'} = 1$ as well as $\gamma(\bar{x})_{i'} = 0$ for the first position $i' \in [n]$ in which both differ. Because γ is ordered, we have $i' \in Z_{\ell'}$. Furthermore, entries of cycles Z_ℓ, $\ell \neq \ell'$ cannot affect the lexicographic order, since they either appear behind $\hat{z}_{\ell'}$ due to the orderedness of the cycles or in front of $\check{z}_{\ell'}$ (and thus in constant cycles). Hence, if x' is the restriction of $\bar{x}$ on $Z_{\ell'}$, we have $\bar{x} \succ \gamma(\bar{x})$ if and only if $x' \succ \zeta_{\ell'}(x')$.

Because γ is monotone and ordered, $\zeta_{\ell'}(x') = (x'_{\hat{z}_{\ell'}}, x'_{\check{z}_{\ell'}}, \dots, x'_{\hat{z}_{\ell'}-1})$. Consequently, $x' \succ \zeta_{\ell'}(x')$ if and only if $x'_{\hat{z}_{\ell'}} = 0$, since x' is not constant along $Z_{\ell'}$. □

Observation 22. Let $\bar{x} \in \mathrm{P}^{\geq}_{\gamma}$ and let $\gamma \in \mathcal{S}_n$ be a monotone and ordered permutation. If cycle ζ_ℓ is a constant cycle of γ w.r.t. $\bar{x}$, then $\bar{x}_i = 1$ for all $i \in Z_\ell$ due to the covering constraints of $\mathrm{P}^{\geq}_{\gamma}$.

In the following, we develop an extended formulation to derive a complete linear description of $\mathrm{P}^{\geq}_{\gamma}$ for monotone and ordered permutations $\gamma \in \mathcal{S}_n$. Due to Lemma 21 and Observation 22, a vertex x of $\mathrm{P}^{\geq}_{\gamma}$ is completely characterized by the index of the first non-constant cycle ℓ', the entries of x on $Z^\star_{\ell'}$, and $\bigcup_{\ell=\ell'+1}^{q} Z_\ell$. Hence, we can completely characterize x by introducing a binary indicator y_ℓ, $\ell \in [q]$, that is 1 if and only if ζ_ℓ is the first non-constant cycle w.r.t. x, as well as a vector $\tilde{x} \in \mathbb{R}^n$ that replaces all entries of x in $\{\hat{z}_{\ell'}\} \cup \bigcup_{\ell=1}^{\ell'-1} Z_\ell$ by 0 and coincides with x otherwise. In particular, we can generate x from the pair $(\tilde{x}, y)$ via the map

$$\pi\colon \mathbb{R}^{n \times q} \to \mathbb{R}^n, \quad \pi(\tilde{x}, y)_i \coloneqq \tilde{x}_i + 1 - \sum_{r=1}^{\ell} y_r, \quad i \in [n],$$

where ℓ is the index of the cycle of γ that contains i. Note that the definition of π, in fact, ensures that the covering constraints on the cycles are fulfilled by $\pi(\tilde{x}, y)$, because $\tilde{x}_i - \sum_{r=1}^{\ell} y_r = 0$ for all $i \in Z_\ell$ with $\ell < \ell'$, and $\sum_{i \in Z_\ell} \tilde{x}_i \geq 1$ as well as $\sum_{r=1}^{\ell} y_r = 1$ for all remaining cycles ℓ.

To derive an extended formulation with the aid of $(\tilde{x}, y)$, we have to enforce that both vectors meet the requirements specified above. To this end, consider the inequalities

$$\sum_{r=1}^{q} y_r \leq 1, \tag{8a}$$

$$\tilde{x}_i \leq \sum_{r=1}^{\ell} y_r, \qquad \ell \in [q],\ i \in Z_\ell^\star, \tag{8b}$$

$$\tilde{x}_{\hat{z}_\ell} \leq \sum_{r=1}^{\ell-1} y_r, \qquad \ell \in [q], \tag{8c}$$

$$\sum_{r=1}^{\ell} y_r \leq \sum_{i \in Z_\ell} \tilde{x}_i, \qquad \ell \in [q], \tag{8d}$$

$$\tilde{x}_i \geq 0, \qquad i \in [n], \tag{8e}$$

$$y_\ell \geq 0, \qquad \ell \in [q]. \tag{8f}$$

Inequality (8a) ensures that y declares at most one cycle as the first non-constant cycle. Inequalities (8b) and (8c) guarantee that $\tilde{x}_i = 0$ if i is contained in a cycle that appears before the first non-constant cycle. Additionally, (8c) enforces that $\tilde{x}_{\hat{z}_\ell} = 0$ if the index of the first non-constant cycle is greater than or equal to ℓ. Finally, Inequality (8d) implies that $\tilde{x}$ fulfills the covering constraint on each cycle that has an index greater or equal than the first non-constant cycle, and Inequalities (8e) and (8f) ensure non-negativity of all variables. Hence, (8) is an IP formulation of the vectors $(\tilde{x}, y)$ that encode a vertex of $\mathrm{P}_\gamma^{\gtrless}$ via π.

Proposition 23. *System (8) describes an integral polytope, and thus, together with the projection π it is an extended formulation of $\mathrm{P}_\gamma^{\gtrless}$.*

Proof. In A, we show that (8) is totally dual integral. Due to the integral right-hand sides in this system, the polytope defined by (8) is integral. Consequently, the projection of (8) via π is $\mathrm{P}_\gamma^{\gtrless}$, which shows the assertion. □

Hence, System (8) not only provides a characterization of tuples $(\tilde{x}, y)$ that encode vertices of $\mathrm{P}_\gamma^{\gtrless}$ via an IP formulation, but it already is a complete linear description of the convex hull of these tuples. To obtain a description of $\mathrm{P}_\gamma^{\gtrless}$ in the original space, we substitute all occurrences of $\tilde{x}_i$ in (8) in accordance with π by

$$\tilde{x}_i = x_i - 1 + \sum_{r=1}^{\ell} y_r,$$

which turns System (8) into

$$\sum_{r=1}^{q} y_r \leq 1, \tag{9a}$$

$$x_i \leq 1, \qquad \ell \in [q],\ i \in Z_\ell^\star, \tag{9b}$$

$$x_{\hat{z}_\ell} + y_\ell \leq 1, \qquad \ell \in [q], \tag{9c}$$

$$-|Z_\ell^\star|\sum_{r=1}^{\ell} y_r - \sum_{i\in Z_\ell} x_i \le -|Z_\ell|, \qquad \ell\in[q], \tag{9d}$$

$$-x_i - \sum_{r=1}^{\ell} y_r \le -1, \qquad \ell\in[q],\ i\in Z_\ell, \tag{9e}$$

$$-y_\ell \le 0, \qquad \ell\in[q]. \tag{9f}$$

The new System (9) consists of two kinds of variables, the original x-variables describing a vector contained in $\mathrm{P}_\gamma^{\geq}$ and the indicator variables y. To obtain a complete linear description of $\mathrm{P}_\gamma^{\geq}$, we project the feasible region of (9) onto the space of x-variables. To determine a description of the projected region, we apply Fourier-Motzkin elimination (FME) to the y-variables. We claim that after the elimination of $y_q,\dots,y_{k+1}$ from (9) the obtained inequality system is

$$\sum_{r=1}^{k} y_r \le 1, \tag{10a}$$

$$x_{\hat{z}_\ell} + y_\ell \le 1, \qquad \ell\in[k], \tag{10b}$$

$$-|Z_\ell^\star|\sum_{r=1}^{\ell} y_r - \sum_{i\in Z_\ell} x_i \le -|Z_\ell|, \qquad \ell\in[k], \tag{10c}$$

$$-x_i - \sum_{r=1}^{\ell} y_r \le -1, \qquad \ell\in[k],\ i\in Z_\ell, \tag{10d}$$

$$-y_l \le 0, \qquad \ell\in[k], \tag{10e}$$

$$-|Z_\ell^\star|\sum_{r=1}^{k} y_r - \sum_{i\in Z_\ell} x_i + |Z_\ell^\star|\sum_{r=k+1}^{\ell} x_{\hat{z}_r} \le b_\ell^k - 1, \qquad \ell\in[q]\setminus[k], \tag{10f}$$

$$-x_i + \sum_{r=k+1}^{\ell} x_{\hat{z}_r} - \sum_{r=1}^{k} y_r \le \ell - k - 1, \qquad \ell\in[q]\setminus[k],\ i\in Z_\ell^\star, \tag{10g}$$

$$-x_i \le 0, \qquad \ell\in[q]\setminus[k],\ i\in Z_\ell, \tag{10h}$$

$$-\sum_{i\in Z_\ell} x_i \le -1, \qquad \ell\in[q]\setminus[k], \tag{10i}$$

$$x_i \le 1, \qquad i\in A^k, \tag{10j}$$

where $A^k := \left(\bigcup_{\ell=1}^{q} Z_\ell^\star\right) \cup \{\hat{z}_j \,:\, j\in\{k+1,\dots,q\}\}$ and $b_\ell^k := |Z_\ell^\star|(\ell-k-1)$.

Lemma 24. *After eliminating variables $y_q,\dots,y_{k+1}$, $k\in[q-1]$, in System (9), the resulting system is given by (10).*

Proof. See A for details. □

Theorem 25. *Let $\gamma\in\mathcal{S}_n$ be a monotone and ordered permutation. Then $\mathrm{P}_\gamma^{\geq}$ is completely described by*

$$-\sum_{i\in Z_\ell} x_i + |Z_\ell^\star|\sum_{r=1}^{\ell} x_{\hat{z}_r} \le |Z_\ell^\star|(\ell-1) - 1, \qquad \ell\in[q],$$

$$-x_i + \sum_{r=1}^{\ell} x_{\hat{z}_r} \le \ell - 1, \qquad \ell\in[q],\ i\in Z_\ell^\star,$$

$$\sum_{i \in Z_\ell} x_i \geq 1, \qquad \ell \in [q],$$
$$0 \leq x_i \leq 1, \qquad i \in [n].$$

Proof. By Lemma 24, a complete linear description of $\mathrm{P}_\gamma^\geq$ is given by System (10) for parameter $k = 0$. Since Inequalities (10a)–(10e) vanish and the remaining inequalities simplify to the proposed constraints, the assertion follows. □

Theorem 25 shows that $\mathrm{P}_\gamma^\geq$ admits a complete linear description of linear size if the underlying permutation is monotone and ordered. Hence, incorporating additional problem information into symmetry handling inequalities allows to handle symmetries by linearly many inequalities with coefficients of linear size. Observe, however, that additional packing or partitioning constraints would have caused that inequalities with ternary coefficients suffice to describe the constrained symresack, see Section 4. This indicates that dealing with additional set covering constraints is more complicated than with set packing or partitioning constraints. In particular, this behavior can also be observed if we consider full orbitopes with additional row-wise set covering constraints, so-called covering orbitopes. While the optimization problem for packing and partitioning orbitopes is polynomial time solvable, the optimization problem over covering orbitopes is NP-hard, see Loos [27].

6. Numerical Experience

To evaluate the effect of the techniques developed in this paper, we have implemented three plug-ins for the framework SCIP [11] to handle symmetries in binary programs: the symresack, orbisack, and orbitope plug-in. Since SCIP version 5.0 these plug-ins are included in SCIP, and thus, publicly available. To make this paper self-contained, we briefly describe these plug-ins below. A detailed description of the three plug-ins can be found in [11]; the theoretical background for methods of the symresack and orbisack plug-ins is provided in [14].

All plug-ins provide methods to separate minimal cover inequalities and/or ordering constraints, however, they differ in the implementation. Moreover, the plug-ins provide a propagation algorithm. The aim of propagation is to detect, at each node v of the branch-and-bound tree, whether there exist variables that are fixed to 0 or 1 in any solution of the subtree rooted in v or to decide that no solution in the subtree rooted in v exist. If such variables are found, the propagation fixes the variables in v to the corresponding values; if infeasibility is detected, the node v is pruned.

The *symresack plug-in* for a symresack P_γ implements a separation routine for minimal cover inequalities of symresacks, which runs in quadratic time. Thus, the separation routine is slower than the theoretically achievable running time of $\mathcal{O}(n\alpha(n))$, cf. Section 2. However, since the running time of this procedure is very small in our experiments, we refrained from implementing the theoretically faster but significantly more complicated method. The idea of the propagation routine is to iterate over all variables x_i in increasing order and to fix variable x_i or $\gamma(x)_i$ to a specific value if this is necessary to ensure $x \succeq \gamma(x)$:

(1) if $x_i = 1$ and $\gamma(x)_i = 0$, terminate: $x \succ \gamma(x)$;
(2) if $x_i = 0$ and $\gamma(x)_i = 1$, terminate: $\gamma(x) \succ x$, prune the current node;
(3) if $x_i = 0$ and $\gamma(x)_i$ is unfixed, fix $\gamma(x)_i = 0$, continue with the next index;
(4) if $\gamma(x)_i = 1$ and x_i is unfixed, fix $x_i = 1$, continue with the next index;
(5) if $x_i = \gamma(x)_i \in \{0, 1\}$, continue with the next index;
(6) otherwise, terminate: no variable fixing can be found.

Figure 2 illustrates this procedure for $\gamma = (1, 4)(2, 5, 7, 8, 3, 6)$ and the variable fixings $x_1 = x_5 = 0$ and $x_3 = x_6 = x_8 = 1$. In this illustration, the first line corresponds to the original solution x and the second line to its permutation $\gamma(x)$.

0		1		0	1		1
	1	1	0		1	0	

(a) Original and permuted partial solution.

0		1	0	0	1		1
0	1	1	0		1	0	

(b) Fixing $\gamma(x)_1 = x_4 = 0$.

0	1	1	0	0	1		1
0	1	1	0	1	1	0	

(c) Fixing $x_2 = \gamma(x)_5 = 1$.

Figure 2. Illustration of the symresack propagation algorithm. Infeasibility is detected in iteration $i = 5$, because $\gamma(x) \succ x$.

Besides these two components, the symresack plug-in is able to check whether the underlying permutation is a composition of 2-cycles. If the check evaluates positively, the symresack can be upgraded to a so-called orbisack, see Kaibel and Loos [27]. These specialized constraints are handled by a second plug-in, the so-called *orbisack plug-in*. In contrast to symresacks, the separation routine for minimal cover inequalities for orbisacks has a linear running time. The propagation routine is the same as for symresacks.

Both the symresack and the orbisack plug-in allow to check whether the cycles of the underlying permutations are constrained by packing or partitioning inequalities. If the symresack plug-in detects that packing/partitioning symresacks are applicable, the ordering constraints (5) are added to the binary program. The orbisack plug-in upgrades itself to an *orbitope plug-in*, the third implemented plug-in, if it is a packing/partitioning orbisack. The orbitope plug-in then separates the ordering constraints and uses a propagation routine that takes packing and partitioning constraints into account. The latter is described in Kaibel et al. [21].

In our experiments, we use a developers' version of SCIP 6.0.2 (git hash a52f4a3) as branch-and-bound framework and CPLEX version 12.9.0 [15] as LP solver. To compute symmetries of the considered instances, we used an internal function of SCIP that computes generators of a subgroup Γ of the symmetry group of a mixed-integer program (MIP) using BLISS 0.73 [18]. The tests were run on a Linux cluster with Intel Xeon E5 3.5GHz quad core processors and 32GB memory; the code was run using one thread and running a single process at a time. The time limit was set to 3600s per instance. All reported average numbers are given in shifted geometric mean

$$\prod_{i=1}^{n} (t_i + s)^{1/n} - s$$

to reduce the impact of very easy instances. We use a shift of $s = 10$ for time and $s = 100$ for the number of nodes. In the tables below, column "#nodes" reports on the average number of nodes needed to solve an instance and column "time" contains the average solution time of each instance. Columns "#opt" and "#act" show how many instances could be solved to optimality within the time limit and in how many instances symmetry handling methods were applied, respectively. Finally, columns "method-time" and "sym-time" report on the average time spent within the symmetry handling routines as well as to detect and initialize the symmetry handling methods, respectively.

6.1. Results for Benchmark Instances

The aim of our experiments on benchmark instances is to investigate the impact of symmetry handling inequalities derived via packing and partitioning symresacks on the performance of a branch-and-bound solver for generic instances. In particular, we are interested in the question whether the additional problem structure of packing or partitioning constraints that is used by ordering constraints (5) improves the general approach via minimal cover inequalities for unconstrained symresacks.

To evaluate the different symmetry handling approaches, we used the following four settings in our experiments. In the `default` setting, all symmetry handling methods of

TABLE 1. Comparison of different symmetry handling variants for general instances.

Setting	#nodes	time	#opt	#act	method-time	sym-time
Miplib2017 (240):						
default	6257.6	890.3	112	0	0.0	0.0
OF	5630.5	837.4	116	75	2.1	0.3
symre	5115.6	859.7	117	96	2.0	0.6
pp-symre	4937.0	846.6	117	96	2.2	0.6
Margot (16):						
default	263 339.1	1183.7	6	0	0.0	0.0
OF	42 634.4	288.1	10	16	1.3	0.0
symre	3849.7	62.4	15	16	0.4	0.0
pp-symre	3051.4	36.4	15	16	0.3	0.0

SCIP are deactivated. The `symre` setting adds for each generator γ of Γ a symresack or orbisack (if γ is a composition of 2-cycles) constraint to the MIP. To handle symmetries of γ, the separation and propagation routines of the corresponding plug-ins are enabled. The upgrade to packing/partitioning symresacks or orbisacks, however, is deactivated in this setting. The `pp-symre` setting extends the `symre` setting by enabling this upgrade. Finally, the `OF` setting uses the state-of-the-art method orbital fixing, see Margot [29, 30] and Ostrowski [35], to handle symmetries, which is also available in SCIP. For the settings `symre` and `pp-symre`, symmetries are computed at the end of SCIP's presolving phase; `OF` computes symmetries after the first branching decision. The timing of symmetry computation differs between the polyhedral approach and orbital fixing, because it is currently not possible to add symmetry handling constraints outside of presolving in SCIP. As orbital fixing is called after the first branching decision, it is favorable not to compute symmetries within presolving for the `OF` setting, to avoid symmetry computation for problems that can be solved within the root node.

Remark 26. We want to stress that in all our experiments described in this and the following section, all generators γ of the symmetry group Γ that allowed to apply packing or partitioning symresacks were monotone. That is, for every instance in which packing/partitioning symresacks are applicable, we can use the facet description of $P^{\leq}_{\gamma}$ and $P^{=}_{\gamma}$. Thus, although the requirement of being monotone is restrictive, it seems that non-monotone packing/partitioning symresacks do not appear in practice—at least when using the generators computed by BLISS.

To be able to test the impact of packing and partitioning symresacks on different kinds of instances, we conducted experiments on two test sets: The 240 instances of the Miplib2017 [34] benchmark test set as well as 16 highly symmetric instances by Margot [29], which are available through the web page [37]. Table 1 provides a summary of our experiments; since the `Margot` test set is rather small, we provide tables with results for each instance in B.

On both the `Miplib2017` and `Margot` test set the time for computing symmetries is below one second on average, which is less than 0.1 % of the average running time. Thus, computing symmetries is not a bottleneck. Note, however, that the time spent for computing symmetries is twice as large for the polyhedral settings `symre` and `pp-symre` as for orbital fixing, which is due to the different timing of computing symmetries. This also explains why orbital fixing is only active on 75 out of 240 `Miplib2017` instances, whereas the polyhedral methods are active on 96 instances: symmetries are always computed in the latter case, while symmetries are not computed for the instances that can be solved within in the root node when using `OF`.

For the `Miplib2017` test set, we observe that `default` SCIP is able to solve 112 instances within the time limit. Symmetry handling improves this number to 115 using `OF` and 116

Table 2. Comparison of different symmetry handling variants on `Miplib2017` instances on which orbital fixing is active.

Setting	#nodes	time	#opt	#act	method-time	sym-time
default	8899.6	1673.3	28	0	0.0	0.0
OF	6334.2	1377.2	32	75	7.1	0.7
symre	4864.5	1425.7	32	71	5.2	0.7
pp-symre	4501.2	1373.4	32	71	5.5	0.7

using the polyhedral settings. Regarding the running time, this leads to an improvement by 5.9 % for orbital fixing as well as 3.4 % for `symre` and 4.9 % for `pp-symre`. That is, although the polyhedral settings are able to solve one more instance than orbital fixing, their average running time is by 1.5 and 1.0 percentage points slower.

To explain this behavior, we evaluated the experiments separately on the 75 instances of `Miplib2017` for which orbital fixing is active, see Table 2. Note that the polyhedral methods are not active on all of these instances as the reductions found within the root node might also introduce symmetries. The experiments reveal that the better performance of `OF` in comparison with `symre` and `pp-symre` is due to the different timings of symmetry computation. When restricting to the symmetric instances that cannot be solved within the root node, `OF` achieves a speed-up of 17.7 %, whereas `symre` improves the performance by 14.5 %. Exploiting the structure of set packing and partitioning constraints, however, allows to improve the performance gain to 18.0 %.

We conclude that there is no great difference between the different symmetry handling methods on the whole test set. On easy instances, handling symmetries harms the solution process as it might destroy the structure making the problem solvable within the root node. On symmetric instances, however, symmetry handling achieves a speed-up of up to 18.0 %, which makes it an important feature of branch-and-bound algorithms. In particular, exploiting the problem structure of set packing and partitioning constraints improves the performance of the polyhedral approach such that it is competitive with orbital fixing. Thus, if a problem contains many set packing or partitioning constraints, ordering inequalities for packing and partitioning symresacks provide an efficient tool for handling symmetries.

The findings for `Miplib2017` are supported by the experiments on the `Margot` test set. These 16 instances are rather hard to solve for SCIP using `default` setting. It solves only 6 instances within the time limit and requires more than 1100 s per instance on average. Since these instances are highly symmetric, it is reasonable that symmetry handling techniques might improve the running time drastically. In particular, since 6 out of 16 instances consist of set packing and partitioning constraints only (the instances whose name begins with “cod”, cf. B), it is likely that packing and partitioning symresacks perform well on these instances: `OR` reduces the running time by 75.6 % and increases the number of solved instances to 10; `symre` allows to reduce the running time by 94.7 % and to solve 15 instances within the time limit. Exploiting additional problem structure, `pp-symre` is even able to reduce the running time of the already fast setting `symre` by further 41.7 %, which leads to an improvement of the `default` setting by 97.0 %. The detailed tables in B show that, indeed, the additional performance gain is achieved on the instances consisting of set packing and partitioning constraints. Here, the running time of `symre` is reduced by 77.3 % on average using the `pp-symre` setting.

In summary, our experiments on benchmark instances show that orbital fixing is a method that reliably improves the performance of branch-and-bound solvers on diverse test sets. Polyhedral methods lead to slightly worse results on average. Depending on the problem structure, however, these methods might outperform orbital fixing as indicated on the `Margot` test set. In particular, if the problems contain many set packing or partitioning constraints, packing and partitioning symresacks are able to exploit this structure to significantly improve on the performance of orbital fixing and unconstrained symresacks.

6.2. Results for Further Symmetric Problems

This section's aim is to support the conclusions found in the last section: If a problem contains many set packing or partitioning constraints, ordering constraints of packing and partitioning symresacks are a simple and powerful tool for handling symmetries. To this end, we considered two different problems classes that contain symmetries that can be handled by packing symresacks: operation room scheduling problems and graph coloring problems.

6.2.1. *Results for Operation Room Scheduling Problems.* The aim of the *operation room scheduling problem (ORSP)*, see Ostrowski et al. [36], is to find a cost minimal assignment of m operations with durations d_i, $i \in [m]$, to n operation rooms in a hospital. A fixed cost f arises if an operation room has to be opened, i.e., there is an operation assigned to this room. Moreover, if the total duration of the operations assigned to a room exceeds a specified time limit T, a cost of v arises for every time unit exceeding T.

By introducing binary variables x_{ij}, $(i, j) \in [m] \times [n]$, and y_j, $j \in [n]$, as well as non-negative continuous variables a_j, $j \in [n]$, ORSP can be modeled as the following mixed-integer program:

$$\min \sum_{j=1}^{n} (f\, y_j + v\, a_j)$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i \in [m], \tag{11a}$$

$$x_{ij} \leq y_j, \qquad (i, j) \in [m] \times [n], \tag{11b}$$

$$\sum_{i=1}^{m} d_i\, x_{ij} \leq T\, y_j + a_j, \qquad j \in [n]. \tag{11c}$$

In this model, $x_{ij} = 1$ if and only if operation i is assigned to room j. Inequality (11a) guarantees that each operation is assigned to exactly one room, and thus, Inequality (11b) ensures that $y_j = 1$ if room j is used in the schedule. Finally, Inequality (11c) models that a_j is (in an optimal solution) the additional time room j has to be opened beyond T. Consequently, the objective measures the total cost of the operation schedule encoded in x.

Since the fixed and variable costs are identical for each room, the room labels of variables in Model (11) can be permuted arbitrarily without changing the problem structure. To illustrate these symmetries of ORSP, we arrange the variables as entries of a matrix

$$X \coloneqq \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \\ y_1 & \dots & y_m \\ a_1 & \dots & a_m \end{pmatrix},$$

which shows that each symmetry corresponds to a reordering of the columns of the variable matrix. Moreover, observe that once the x-variables are fixed, the optimal values of the y- and a-variables can be determined easily. Thus, it suffices in principle to handle only symmetries on the x-variables to handle all room symmetries of ORSP.

Intuitively, all room symmetries can be handled by enforcing that the columns of the x-matrix are sorted lexicographically non-increasing. One way to achieve this is to add

Table 3. Comparison of different symmetry handling variants for `OR` instances. The different test sets are encoded by the parameters m, n, T as `OR_m_n_T`.

Setting	#nodes	time	#opt	#act	method-time	sym-time
`OR_30_20_120`:						
default	45 466.6	457.7	19	0	0.0	0.0
OF	41 270.7	341.4	25	29	11.0	0.0
symre	86.6	10.0	50	50	0.0	0.0
prob. spec.	207.9	8.3	50	50	0.0	0.0
`OR_40_20_140`:						
default	177 834.5	795.5	17	0	0.0	0.0
OF	169 733.0	765.4	18	50	21.5	0.0
symre	39 956.1	273.9	41	50	3.5	0.0
prob. spec.	22 484.0	226.7	41	50	0.0	0.0
`OR_60_30_140`:						
default	542 990.2	1963.2	13	0	0.0	0.0
OF	621 012.3	2257.4	12	50	76.5	0.0
symre	124 448.4	1230.0	20	50	13.6	0.1
prob. spec.	94 212.8	973.2	17	50	0.0	0.0

the constraint that the x-matrix is contained in a partitioning orbitope, cf. Section 1. Alternatively, one can add symresack constraints for the permutations γ_j, $j \in [n-1]$, which swap the corresponding entries of columns j and $j+1$ of the variable matrix, i.e.,

$$\gamma_j = (x_{1j}, x_{1(j+1)})(x_{2j}, x_{2(j+1)}) \dots (x_{mj}, x_{m(j+1)}),$$

see [14, Corollary 31]. Consequently, because $x_{i,j} + x_{i(j+1)} \leq 1$ by Inequality (11a) and non-negativity of x, the symmetries γ_j can be handled by packing symresacks, and thus, by the linearly many Inequalities (5).

Observe, however, that a general MIP solver might not be able to detect that partitioning orbitopes or packing symresacks are applicable because symmetries also affect the y- and a-variables which are not constrained by packing or partitioning constraints. Of course, the MIP solver could restrict the permutations γ to the cycles that are constrained by packing inequalities to be able to use packing symresacks. However, this approach ignores the symmetry information of unconstrained cycles and it is unclear whether this is beneficial in a general MIP.

The goal of this section is to compare these two families of approaches, i.e., approaches that use the full symmetry information and approaches that exploit only the symmetries on the x-variables. For our experiments, we generated for each parameter setting $(m, n, T) \in \{(30, 20, 120), (40, 20, 140), (60, 30, 140)\}$ 50 random instances of ORSP to compare the performance of the different approaches over instances of varying difficulty.[1] The used settings are `default`, `OF`, and `symre`—we do not report on results for `pp-symre`, because SCIP does not (yet) autonomously detect that it suffices to handle only symmetries of x-variables. Hence, the results for `symre` and `pp-symre` would be the same. To be able to investigate the impact of packing symresacks on ORSP, we add the ordering constraints (5) directly to the problem formulation. We refer to this setting as the problem specific approach (`prob. spec.`). The results of our experiments are summarized in Table 3.

For an increasing number of operations m, the problems become harder. For this reason, we refer to the instances with $(m, n, T) = (30, 20, 120)$ as easy instances and to $(40, 20, 140)$ and $(60, 30, 140)$ as medium and hard instances, respectively. Throughout all three test sets, we observe that handling symmetries is important to solve these instances efficiently, where the reduction of solution time decreases with increasing difficulty. This, however,

[1]The instances are available through the author's web page https://www.win.tue.nl/~chojny/.

can also be a secondary effect, because less of the hard instances than of the easy instances can be solved within the time limit.

On all test sets, the best tested settings are the polyhedral approaches `symre` and `prob. spec.`, where the problem specific approach leads always to the best improvement in running time. While already the easy instances are challenging for SCIP as it solves only 19 of the 50 instances, both polyhedral methods allow to solve all instances within the time limit leading to a reduction of solving time by 97.8 % and 98.2 %, respectively. On the medium and hard instances, not all of the instances can be solved to optimality. But for the medium instances still 24 additional instances can be solved, reducing the solution time by 65.6 % and 71.5 %, respectively. For the hard instances, we observe that the problem specific approach is still able to achieve the best reduction of running time, 50.4 % in comparison to 37.3 % for `symre`. The `symre` setting, however, is able to solve seven additional instances, say I_s, whereas the problem specific approach improves the number of solved instances only by four, say I_p. The difference set $I_p \setminus I_s$ contains one instance, i.e., there exists an instance that can be solved by the problem specific approach that is unsolvable for setting `symre`. Comparing the results for the affected instances, shows that SCIP always found an optimal solution for the instances in I_s using the problem specific approach. However, it was not able to prove optimality. This is remarkable as one might have expected that the formulation using ordering constraints is much more restrictive, making it harder to find good solutions. The final optimal solution, was always found due to an integral LP solution. Thus, the well-structured ordering inequalities with ternary coefficient seem to enforce integrality and not to improve the final dual bound. In contrast to this, for the single instance in $I_p \setminus I_s$, SCIP was able to find the correct dual bound, but not an optimal solution.

Orbital fixing is hardly able to improve the number of solved instances in comparison with the `default` setting, although the time spent for performing the reduction steps of orbital fixing is much larger than the sym-time of `symre`. As orbital fixing is a symmetry handling method for general binary programs, these experiments suggest that exploiting additional problem structure may drastically improve the performance of symmetry handling methods. Note that the performance gain of the `symre` setting is still very high although it does not explicitly take the packing and partitioning structure into account. This indicates that generic symmetry handling inequalities interact with this problem structure, leading to a better performance than orbital fixing which does not add inequalities to the problem. Explicitly exploiting packing and partitioning constraints using packing and partitioning symresacks, however, leads to a better performance improvement than using unconstrained symresacks, which makes it worth it to check whether constrained symresacks are applicable for a concrete problem. These conclusions are also supported by our findings for the `Margot` test set.

6.2.2. *Results for Star Coloring Problems.* In this section, we report on the effect of symmetry handling for a variant of the maximum 2-colorable subgraph problem, the maximum 2-star colorable subgraph problem (M2SCP). This problem extends M2CP by adding the additional requirement that every path on 4 nodes in the underlying graph is not bicolored, i.e., we impose the additional constraints

$$\sum_{v \in P} (x_{v1} + x_{v2}) \leq 3, \qquad P \in \mathcal{P},$$

where $\mathcal{P}$ denotes the set of all path subgraphs containing four nodes, to the IP formulation of M2CP in Example 9. Star colorings arise, for example, in the efficient computation of sparse Hessian matrices, see, e.g., Gebremedhin et al. [8, 9]

As M2CP, M2SCP contains two kinds of symmetries: graph and color symmetries. While color symmetries can easily be handled by packing orbitopes, the derivation of symmetry

Table 4. Comparison of different symmetry handling variants for star coloring instances.

Setting	#nodes	time	#opt	#act	method-time	sym-time
Color02 (81):						
default	10 286.0	360.41	38	0	0.00	0.00
partsym	9867.7	351.00	38	81	1.56	0.00
sym	9490.6	340.84	42	81	3.02	0.00
pp-symre	7607.5	304.98	42	81	2.63	0.00
Color02-clique (51):						
default	8297.9	194.99	29	0	0.00	0.00
partsym	8560.4	195.87	29	51	2.09	0.00
sym	8592.5	196.60	33	51	3.79	0.01
pp-symre	6040.8	161.97	33	51	3.16	0.01

handling inequalities for graph symmetries is typically more complicated and tedious, see, e.g., Januschowski and Pfetsch [16, 17]. The framework of symresacks, however, allows to easily find symmetry handling inequalities for both color and graph symmetries. In particular, if a graph automorphism acts on a clique, packing symresacks can be used to handle the automorphism, see Example 9. The aim of this section is to investigate whether graph symmetries of cliques can efficiently be handled by packing symresacks.

In our experiments for M2SCP, we extended the star coloring solver described in [13] by a routine to handle graph symmetries. This method uses NAUTY [33] to detect graph automorphisms and it checks whether the automorphism acts on a clique. To test the impact of the different variants to handle graph symmetries in M2SCP, we use the following settings: In the `default` setting, no symmetries are handled, whereas the setting `colsym` uses a packing orbitope to handle color symmetries. The setting `sym` extends `colsym` by handling graph automorphisms by unconstrained symresacks; the last setting `pp-sym` extends `sym` by checking whether a graph automorphism acts on a clique. If a clique automorphism is detected, a packing symresack is used to handle the graph automorphism; if it does not act on a clique, an unconstrained symresack is used. We do not compare the above methods with orbital fixing, because the star coloring solver extends SCIP by problem specific plug-ins, which prevent SCIP to apply orbital fixing.

To test the effect of the different settings, we ran experiments on instances from the `Color02` symposium [5]. Since memory consumption might be very large if the graphs are big, we only considered the 81 graph instances for which we were able to free the memory allocated within the solution process in less 70 minutes. Moreover, we considered the subset `Color02-clique` of instances in `Color02` which contain automorphisms of cliques, i.e., every instance in `Color02-clique` contains a graph symmetry that can be handled by a packing symresack. The results of our experiments are provided in Table 4.

On the whole test set `Color02`, handling color symmetries leads only to a minor improvement of the running time by 2.6 %. Additionally handling graph symmetries is more important as it allows to solve four additional instances and to improve the running time by 5.4 %. Exploiting the additional packing structure of clique automorphisms, however, leads to a running time reduction of 15.4 %. Restricting to the instances of `Color02-clique`, handling symmetries but not exploiting clique structures is almost performance neutral. Handling automorphisms of cliques by packing symresacks, reduces the running time by 16.9 %. Since detecting graph automorphisms and checking whether they act on cliques is not costly on the tested instances, this again shows that incorporating problem structure into symmetry handling inequalities leads to a significantly better performance than using general purpose inequalities.

7. Conclusion

In this paper, we have presented a generalization of the symresack framework for deriving symmetry handling inequalities to constrained symresacks, which allows to derive symmetry handling cutting planes that incorporate additional problem structure. By showing that linear optimization problems over cardinality constrained symresacks can be solved in cubic time, we provided theoretical evidence that symmetry handling inequalities incorporating packing, partitioning, or covering constraints can be separated in polynomial time. To obtain control on the kind of inequalities we separate and to get a better bound on the separation complexity, we investigated the special case of packing, partitioning, and covering symresacks in more detail. We proved that these polytopes admit small formulations with bounded coefficients in certain cases. In practice, we have seen that the general framework of constrained symresacks allows to derive symmetry handling inequalities that are essential to solve symmetric problems containing packing and partitioning structures efficiently.

To further improve the performance of a MIP solver on general instances, it is important to identify additional problem structures that can efficiently be incorporated into symmetry handling inequalities extending the family of constrained symresacks. Furthermore, the complete linear descriptions of constrained symresacks we were able to derive are only valid for monotone permutations. Another direction of research is to drop this requirement and to investigate the corresponding constrained symresacks. Both may have a positive impact on the performance of a MIP solver, but are out of scope of this paper.

Acknowledgments. The author thanks Tristan Gally, Oliver Habeck, Marc E. Pfetsch as well as four anonymous referees for their helpful comments on an earlier version of this paper, which helped to improve the presentation and quality of this article. Moreover, the author thanks Marc E. Pfetsch for many fruitful discussions.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix A. Missing Proofs

In this section, we provide the proofs missing in the main part of this article.

Theorem 15. *Let $\gamma \in \mathcal{S}_n$ be a monotone permutation without fixed points. Then $\mathrm{P}^{=}_{\gamma}$ is completely described by* (7).

Proof. Let $P = \mathrm{conv}(\{x \in \mathbb{R}^n : x \text{ fulfills (7a)–(7c)}\})$ and abbreviate the defining inequality/equation system of P with $Ax \,(\leqq)\, b$. Note that $\mathrm{P}^{=}_{\gamma} = \mathrm{conv}(P \cap \mathbb{Z}^n)$ by Proposition 12. Hence, if we can establish that each vertex of P is integral, we have proven that $P = \mathrm{P}^{=}_{\gamma}$.

A basis $\mathbb{B}$ of a vertex $\bar{x}$ of P is a set of n constraints from (7a)–(7c) whose coefficient vectors of the left-hand sides form a regular matrix $A_{\mathbb{B}}$ such that $\bar{x}$ is the unique solution of $A_{\mathbb{B}}x = b_{\mathbb{B}}$, where the subscript $\mathbb{B}$ denotes the restriction of A and b to the constraints in $\mathbb{B}$. To show integrality of $\bar{x}$, we prove some structural properties of reduced bases, which will be defined below. These structural properties will allow us to conclude that each vertex of P has to be integral. This proof technique was already used by Kaibel and Pfetsch [20] to show validity of a complete linear description of partitioning orbitopes.

Given a vertex $\bar{x}$ of P, we call a basis $\mathbb{B}$ of $\bar{x}$ *reduced* if it contains all active non-negativity constraints (7c), i.e., the non-negativity constraints $-x_i \leq 0$ for all $i \in [n]$ such that $\bar{x}_i = 0$, and all partitioning constraints (7b). Note that reduced bases exist for each vertex $\bar{x}$ of P because the coefficient vectors of the contained partitioning and non-negativity constraints are linearly independent.

To prove integrality of each vertex $\bar{x}$ of P, we show that $\bar{x}$ has a (reduced) basis $\mathbb{B}$ which does not contain any ordering constraint. Consequently, $\mathbb{B}$ contains q partitioning constraints and $n - q$ non-negativity constraints, and thus, $\bar{x}$ has at least $n - q$ non-zero entries. But since $\bar{x}$ fulfills all partitioning constraints, each cycle support Z_ℓ contains at least one non-zero entry. This proves that $\bar{x}$ contains exactly q ones and $n - q$ zeros, and hence, is binary.

For this reason, it suffices to prove that each vertex of P admits a basis that does not contain any ordering constraint.

Claim 27. *A reduced basis of a vertex $\bar{x}$ of P does not contain any ordering constraint.*

To prove this claim, we proceed as follows. First, we argue that a reduced basis of $\bar{x}$ cannot contain some special ordering constraints. Second, we prove for each support Z_ℓ a structural property on the distribution of non-zero entries $\bar{x}_i$, $i \in Z_\ell$. Finally, we exploit the first two steps to show that $\bar{x}$ cannot be the unique solution of $A_{\mathbb{B}}x = b_{\mathbb{B}}$ if $\mathbb{B}$ is a reduced basis that contains an ordering constraint. This allows to deduce that a reduced basis of $\bar{x}$ cannot contain any ordering constraint, and thus, $\bar{x}$ is integral by the above argument.

Let $\bar{x}$ be a vertex of P and let $\mathbb{B}$ be a reduced basis that contains an ordering constraint for $\ell \in [q]$. We call this constraint *trivial* if either $\sum_{i=1}^{\check{z}_\ell - 1} \bar{x}_i = \bar{x}_{\hat{z}_\ell} = 0$ or $\bar{x}_{\hat{z}_\ell} = 1$ and there exists exactly one $j \in [\check{z}_\ell - 1]$ such that $\bar{x}_j = 1$. In the latter case, $\bar{x}_i = 0$ holds for all $i \in [\check{z}_\ell - 1] \setminus \{j\}$ since the ordering constraint is contained in $\mathbb{B}$.

Claim 28. *A reduced basis $\mathbb{B}$ of $\bar{x}$ does not contain any trivial ordering constraint.*

Proof. Assume there exists a trivial ordering constraint that is contained in a reduced basis $\mathbb{B}$ of $\bar{x}$. Denote with $\ell \in [q]$ the index of this trivial ordering constraint.

If $\sum_{i=1}^{\check{z}_\ell - 1} \bar{x}_i = \bar{x}_{\hat{z}_\ell} = 0$, the non-negativity constraints $-x_i \leq 0$ are fulfilled by $\bar{x}$ with equality for each $i \in [\check{z}_\ell - 1] \cup \{\hat{z}_\ell\}$. Since $\mathbb{B}$ is a reduced basis, all these non-negativity constraints are contained in $\mathbb{B}$. Thus, we can generate the left-hand side of the ordering constraint with index ℓ by adding all non-negativity constraints for $i \in [\check{z}_\ell - 1]$ to the negative of the non-negativity constraint $-x_{\hat{z}_\ell} \leq 0$. Consequently, $A_{\mathbb{B}}$ cannot be regular, which is a contradiction to $\mathbb{B}$ being a basis.

Otherwise, $\bar{x}_{\hat{z}_\ell} = 1$ and there exists exactly one $j \in [\check{z}_\ell - 1]$ such that $\bar{x}_j = 1$. Let ℓ' be the index of the cycle containing j. Then the non-negativity constraints and partitioning constraints on cycles Z_ℓ and $Z_{\ell'}$ imply that $\bar{x}_i = 0$ holds for every $i \in (Z_{\ell'} \cup Z_\ell) \setminus \{j, \hat{z}_\ell\}$. But since $\mathbb{B}$ is reduced and the ordering constraint for ℓ is trivial, $\mathbb{B}$ contains the non-negativity constraints for all $i \in ([\check{z}_\ell - 1] \cup Z_{\ell'} \cup Z_\ell) \setminus \{j, \hat{z}_\ell\}$. Again, we obtain a contradiction to $\mathbb{B}$ being a basis because we can generate the left-hand side of the ordering constraint for ℓ by the following linear combination of left-hand sides of constraints in $\mathbb{B}$, where the first two sums correspond to partitioning constraints and the remaining sums correspond to non-negativity constraints:

$$\sum_{i \in Z_\ell} x_i - \sum_{i \in Z_{\ell'}} x_i + \sum_{i \in [\check{z}_\ell - 1] \setminus \{j\}} (-x_i) + \sum_{i \in Z_\ell^\star} (-x_i) - \sum_{i \in Z_{\ell'} \setminus \{j\}} (-x_i) = -\sum_{i=1}^{\check{z}_\ell - 1} x_i + x_{\hat{z}_\ell}.$$

$\diamondsuit$

Next, we prove some structural properties of the distribution of non-zero entries of $\bar{x}$ on each support Z_ℓ. These properties will be crucial to prove Claim 27. Before we proceed with the first property described in Claim 29, we group the ordering constraints of a basis into two classes: If the ordering constraint for $\ell \in [q]$ is fulfilled by $\bar{x}$ with equality and $\bar{x}_{\hat{z}_\ell} = 1$, we call the ordering constraint 1-*active*. Otherwise, if the ordering constraint is fulfilled with equality but $\bar{x}_{\hat{z}_\ell} < 1$, it is called *fractionally active*.

Claim 29. *Let $\mathbb{B}$ be a reduced basis of $\bar{x}$ that does not contain any 1-active ordering constraint. Then for each $\ell \in [q]$ there exists an index $j \in Z_\ell$ that is not contained in the support of an ordering constraint in $\mathbb{B}$ such that $\bar{x}_j > 0$.*

Proof. If $\bar{x}_{\hat{z}_\ell} = 1$ for any $\ell \in [q]$, the ordering constraint for ℓ cannot be contained in $\mathbb{B}$ since $\mathbb{B}$ does not contain any 1-active ordering constraint. Moreover, $\hat{z}_\ell$ cannot be contained in any ordering constraint in $\mathbb{B}$, because this would imply that this other ordering constraint would be 1-active as well. Thus, there exists $j \in Z_\ell$, namely $j = \hat{z}_\ell$, such that $\bar{x}_j > 0$ and j is not contained in the support of any ordering constraint in $\mathbb{B}$. Consequently, it suffices to consider only those cycles $\ell' \in [q]$ for which $\bar{x}_{\hat{z}_{\ell'}} < 1$.

If the claim was false, there would be $\ell' \in [q]$ with $\bar{x}_{\hat{z}_{\ell'}} < 1$ such that each $j \in Z_{\ell'}$ with $\bar{x}_j > 0$ is contained in the support of a fractionally active ordering constraint. Denote with $\hat{\jmath}$ the greatest index in $Z_{\ell'}^\star$ with $\bar{x}_{\hat{\jmath}} > 0$, and let $\tilde{\ell}$ be the index of a cycle with $\hat{\jmath} < \check{z}_{\tilde{\ell}}$ and whose ordering constraint is contained in $\mathbb{B}$. The indices $\hat{\jmath}$ and $\tilde{\ell}$ are well-defined because $\sum_{j \in Z_{\ell'}} \bar{x}_j = 1$ and $\bar{x}_{\hat{z}_{\ell'}} < 1$ as well as all positive variables in $Z_{\ell'}$ are contained in the support of an active ordering constraint.

By definition of $\hat{\jmath}$ and $\tilde{\ell}$ as well as by the partitioning constraints, we have

$$\bar{x}_{\hat{z}_{\ell'}} + \sum_{i \in [\check{z}_{\tilde{\ell}} - 1] \cap Z_{\ell'}^\star} \bar{x}_i = 1 \tag{12}$$

and $\check{z}_{\ell'} \leq \hat{\jmath} < \check{z}_{\tilde{\ell}}$.

For the fractionally active ordering constraint $\tilde{\ell}$, we can estimate

$$\begin{aligned} 1 \geq \bar{x}_{\hat{z}_{\tilde{\ell}}} &= \sum_{i=1}^{\check{z}_{\tilde{\ell}} - 1} \bar{x}_i = \sum_{i=1}^{\check{z}_{\ell'} - 1} \bar{x}_i + \sum_{i = \check{z}_{\ell'}}^{\check{z}_{\tilde{\ell}} - 1} \bar{x}_i \geq \sum_{i=1}^{\check{z}_{\ell'} - 1} \bar{x}_i + \sum_{i \in [\check{z}_{\tilde{\ell}} - 1] \cap Z_{\ell'}^\star} \bar{x}_i \\ &\overset{(12)}{=} \sum_{i=1}^{\check{z}_{\ell'} - 1} \bar{x}_i + 1 - \bar{x}_{\hat{z}_{\ell'}} \overset{(7a)}{\geq} \bar{x}_{\hat{z}_{\ell'}} + 1 - \bar{x}_{\hat{z}_{\ell'}} = 1. \end{aligned}$$

Hence, equality holds throughout this inequality chain which implies that the ordering constraint for $\tilde{\ell}$ is 1-active. This is a contradiction to the assumption on $\mathbb{B}$.

Consequently, for each $\ell \in [q]$ there exists an index $j \in Z_\ell$ with $\bar{x}_j > 0$ such that j is not contained in the support of any ordering constraint in $\mathbb{B}$. ◊

Claim 30. *A reduced basis contains at most one* 1*-active ordering constraint.*

Proof. Assume there exists a reduced basis $\mathbb{B}$ that contains two 1-active ordering constraints ℓ_1 and ℓ_2, where $\check{z}_{\ell_1} < \check{z}_{\ell_2}$. Then $\bar{x}_{\hat{z}_{\ell_1}} = \bar{x}_{\hat{z}_{\ell_2}} = 1$ as well as $\bar{x}_i = 0$ for all $i \in Z^\star_{\ell_1} \cup Z^\star_{\ell_2} =: I_1 \cup I_2$. Moreover, 1-activeness of the ordering constraint for ℓ_1 and $\check{z}_{\ell_1} < \check{z}_{\ell_2}$ implies that $\bar{x}_i = 0$ for all $i \in I_3 := \{\check{z}_{\ell_1}, \dots, \check{z}_{\ell_2} - 1\}$, since otherwise, the ordering constraint for ℓ_2 cannot be fulfilled with equality. Thus, $\bar{x}_i = 0$ for all $i \in I := I_1 \cup I_2 \cup I_3$.

Because $\mathbb{B}$ is a reduced basis of $\bar{x}$, $\mathbb{B}$ contains the non-negativity constraints for all indices in I. Thus, we can generate the left-hand side of the ordering constraint of ℓ_2 by the following linear combination of non-negativity constraints for variables $\bar{x}_i$, $i \in I$, the ordering constraint for ℓ_1 and the partitioning constraints for ℓ_1 and ℓ_2:

$$\Big(-\sum_{i=1}^{\check{z}_{\ell_1}-1} x_i + x_{\hat{z}_{\ell_1}}\Big) - \sum_{i \in Z_{\ell_1}} x_i - \sum_{i \in Z^\star_{\ell_1}} (-x_i) + \sum_{i \in I_3} (-x_i)$$

$$- \sum_{i \in Z_{\ell_2}} (-x_i) + \sum_{i \in Z^\star_{\ell_2}} (-x_i) = -\sum_{i=1}^{\check{z}_{\ell_2}-1} x_i + x_{\hat{z}_{\ell_2}}.$$

Since $A_{\mathbb{B}}$ has to be regular, this is a contradiction. ◊

Claim 31. *Let* $\mathbb{B}$ *be a reduced basis of* $\bar{x}$ *that contains exactly one* 1*-active ordering constraint* $\ell' \in [q]$*. Then there is at most one* $\tilde{\ell} \in [q]$*,* $\tilde{\ell} \neq \ell'$*, such that each* $j \in Z_{\tilde{\ell}}$ *with* $\bar{x}_j > 0$ *is contained in the support of an ordering constraint in* $\mathbb{B}$*.*

Proof. First, assume there exists $\tilde{\ell} \in [q]$ such that

$$\sum_{i \in Z_{\tilde{\ell}} \cap [\check{z}_{\ell'}-1]} \bar{x}_i = 1,$$

i.e., each $j \in Z_{\tilde{\ell}} \cap \operatorname{supp}(\bar{x})$ is contained in the support of the ordering constraint of ℓ'. Then

$$\sum_{i \in Z_\ell \cap [\check{z}_{\ell'}-1]} \bar{x}_i = 0 \tag{13}$$

for each $\ell \in [q] \setminus \{\ell', \tilde{\ell}\}$, since otherwise, the ordering constraint for ℓ' cannot be 1-active. This implies that there exists $j \in Z_\ell \setminus [\check{z}_{\ell'} - 1]$ with $\bar{x}_j > 0$ for every $\ell \in [q] \setminus \{\ell', \tilde{\ell}\}$ because of the partitioning constraints. Now, distinguish the following two cases:

On the one hand, assume $\bar{x}_{\hat{z}_\ell} < 1$. Then there exists $j \in Z^\star_\ell \setminus [\check{z}_{\ell'} - 1]$ with $\bar{x}_j > 0$ due to the partitioning constraint for ℓ. Since j is not contained in the support of the ordering constraint of ℓ', the index j cannot be contained in the support of any ordering constraint in $\mathbb{B}$. On the other hand, consider $\bar{x}_{\hat{z}_\ell} = 1$. If $\hat{z}_\ell$ is contained in the support of an ordering constraint in $\mathbb{B}$, then this ordering constraint has to be 1-active. But since the only 1-active ordering constraint in $\mathbb{B}$ is the ordering constraint of ℓ' and $\hat{z}_\ell > \check{z}_{\ell'} - 1$ by (13), $\hat{z}_\ell$ cannot be contained in the support of the ordering constraint of ℓ'. Consequently, $\hat{z}_\ell$ is not contained in the support of any ordering constraint in $\mathbb{B}$. For this reason, Z_ℓ contains in both cases an element that is not contained in the support of an ordering constraint in $\mathbb{B}$, which proves that there is at most one $\tilde{\ell} \in [q] \setminus \{\ell'\}$ such that every $j \in Z_{\tilde{\ell}}$ with $\bar{x}_j > 0$ is contained in the support of an ordering constraint in $\mathbb{B}$.

As a consequence, we can assume in the following that

$$\sum_{i \in Z_\ell \cap [\check{z}_{\ell'}-1]} \bar{x}_i < 1$$

holds for each $\ell \in [q]$. If there exists a cycle $\tilde{\ell} \in [q] \setminus \{\ell'\}$ such that each $j \in Z_{\tilde{\ell}}$ with $\bar{x}_j > 0$ is contained in the support of an ordering constraint in $\mathbb{B}$, we have

$$\bar{x}_{\hat{z}_{\tilde{\ell}}} + \sum_{i \in Z_{\tilde{\ell}} \cap [\check{z}_{\ell'} - 1]} \bar{x}_i = 1 : \tag{14}$$

Indeed, assume there exists $j \in Z^{\star}_{\tilde{\ell}} \setminus [\check{z}_{\ell'} - 1]$ such that $\bar{x}_j > 0$. Since every positive entry of $\bar{x}$ on $Z_{\tilde{\ell}}$ is contained in the support of an ordering constraint in $\mathbb{B}$, there exists $\ell'' \in [q]$ whose ordering constraint is contained in $\mathbb{B}$ and that fulfills $\check{z}_{\ell'} < \check{z}_{\ell''}$. But then the ordering constraint for ℓ'' cannot be contained in $\mathbb{B}$ because

$$\sum_{i=1}^{\check{z}_{\ell''}-1} \bar{x}_i \geq \bar{x}_j + \sum_{i=1}^{\check{z}_{\ell'}-1} \bar{x}_i > \sum_{i=1}^{\check{z}_{\ell'}-1} \bar{x}_i = 1,$$

a contradiction. Consequently, the only positive entry of $\bar{x}$ on $Z_{\tilde{\ell}} \setminus [\check{z}_{\ell'} - 1]$ can be $\hat{z}_{\tilde{\ell}}$. In particular, since $\hat{z}_{\tilde{\ell}}$ has to be contained in the support of an ordering constraint in $\mathbb{B}$ and $\sum_{i \in Z_{\tilde{\ell}} \cap [\check{z}_{\ell'} - 1]} \bar{x}_i < 1$ holds, this implies that the ordering constraint for $\tilde{\ell}$ is contained in $\mathbb{B}$.

If $\bar{x}_{\hat{z}_{\tilde{\ell}}} = 1$, the ordering constraint for $\tilde{\ell}$ cannot be contained in $\mathbb{B}$ by assumption. Thus, we can assume that $\bar{x}_{\hat{z}_{\tilde{\ell}}} \in (0, 1)$, and, in particular, (14) then implies $\check{z}_{\tilde{\ell}} < \check{z}_{\ell'}$. Since these arguments hold for each $\tilde{\ell} \in [q]$ that fulfills (14), assume for the sake of contradiction that there exist $\tilde{\ell}_1, \tilde{\ell}_2 \in [q]$, $\check{z}_{\tilde{\ell}_1} < \check{z}_{\tilde{\ell}_2} < \check{z}_{\ell'}$, such that both fulfill (14). With these assumptions, we have

$$\begin{aligned} 1 = \sum_{i=1}^{\check{z}_{\ell'}-1} \bar{x}_i &\geq \sum_{i=1}^{\check{z}_{\tilde{\ell}_1}-1} \bar{x}_i + \sum_{i \in Z_{\tilde{\ell}_1} \cap [\check{z}_{\ell'}-1]} \bar{x}_i + \sum_{i \in Z_{\tilde{\ell}_2} \cap [\check{z}_{\ell'}-1]} \bar{x}_i \\ &\overset{(7a)}{\geq} \bar{x}_{\hat{z}_{\tilde{\ell}_1}} + \sum_{i \in Z_{\tilde{\ell}_1} \cap [\check{z}_{\ell'}-1]} \bar{x}_i + \sum_{i \in Z_{\tilde{\ell}_2} \cap [\check{z}_{\ell'}-1]} \bar{x}_i \\ &\overset{(14)}{=} \bar{x}_{\hat{z}_{\tilde{\ell}_1}} + 1 - \bar{x}_{\hat{z}_{\tilde{\ell}_1}} + 1 - \bar{x}_{\hat{z}_{\tilde{\ell}_2}} = 2 - \bar{x}_{\hat{z}_{\tilde{\ell}_2}} > 1. \end{aligned}$$

Because this inequality chain cannot be valid, there exists at most one $\tilde{\ell} \in [q]$ which fulfills (14). Thus, the claim holds. ◊

Claim 32. *Let $\mathbb{B}$ be a reduced basis of $\bar{x}$ that contains exactly one 1-active ordering constraint $\ell' \in [q]$ and let $\hat{\imath}$ be the greatest index in $[\check{z}_{\ell'} - 1]$ with $\bar{x}_{\hat{\imath}} > 0$. If there exists $\tilde{\ell} \in [q]$, $\tilde{\ell} \neq \ell'$, such that each $j \in Z_{\tilde{\ell}}$ with $\bar{x}_j > 0$ is contained in the support of an ordering constraint in $\mathbb{B}$, then $\hat{\imath} \in Z^{\star}_{\tilde{\ell}}$.*

Proof. Let $\tilde{\ell} \in [q]$ such that each $i \in Z_{\tilde{\ell}}$ with $\bar{x}_j > 0$ is contained in the support of an ordering constraint in $\mathbb{B}$. If $\hat{z}_{\tilde{\ell}} < \check{z}_{\ell'}$, we have

$$1 = \bar{x}_{\hat{z}_{\ell'}} = \sum_{i=1}^{\check{z}_{\ell'}-1} \bar{x}_i \geq \sum_{i=1}^{\hat{z}_{\tilde{\ell}}} \bar{x}_i \geq \sum_{i \in Z_{\tilde{\ell}}} \bar{x}_i = 1.$$

Hence, $\hat{\imath} \in Z_{\tilde{\ell}}$ and $\bar{x}_j = 0$ for all $j \in Z_\ell \cap [\check{z}_{\ell'} - 1]$, $\ell \in [q] \setminus \{\tilde{\ell}\}$. Consequently, $\sum_{i=1}^{\check{z}_{\tilde{\ell}}-1} \bar{x}_i = 0$ holds. Thus, $\bar{x}_{\hat{z}_{\tilde{\ell}}} = 0$ by (7a), which implies $\hat{\imath} \in Z^{\star}_{\tilde{\ell}}$.

As a consequence, we can assume that $\hat{z}_{\tilde{\ell}} > \check{z}_{\ell'}$. Since every positive entry of $\bar{x}$ on $Z_{\tilde{\ell}}$ is contained in the support of an ordering constraint in $\mathbb{B}$ and ℓ' is the only 1-active ordering constraint in $\mathbb{B}$, we have

$$\bar{x}_{\hat{z}_{\tilde{\ell}}} + \sum_{i \in Z_{\tilde{\ell}} \cap [\check{z}_{\ell'}-1]} \bar{x}_i = 1.$$

This implies that each $j \in Z^\star_{\tilde{\ell}}$ with $\bar{x}_j > 0$ is contained in the support of the ordering constraint for ℓ'. Then,

$$1 = \bar{x}_{\hat{z}_{\ell'}} = \sum_{i=1}^{\check{z}_{\ell'}-1} \bar{x}_i \geq \sum_{i=1}^{\check{z}_{\tilde{\ell}}-1} \bar{x}_i + \sum_{i \in Z_{\tilde{\ell}} \cap [\check{z}_{\ell'}-1]} \bar{x}_i \overset{(7a)}{\geq} \bar{x}_{\hat{z}_{\tilde{\ell}}} + \sum_{i \in Z_{\tilde{\ell}} \cap [\check{z}_{\ell'}-1]} \bar{x}_i = 1.$$

For this reason, $\bar{x}_j = 0$ for each $j \in \{\check{z}_{\tilde{\ell}}, \dots, \check{z}_{\ell'} - 1\} \setminus Z_{\tilde{\ell}}$. Moreover, $\bar{x}_{\hat{z}_{\tilde{\ell}}} < 1$, because otherwise, the ordering constraint for $\tilde{\ell}$ has to be 1-active and it has to be contained in $\mathbb{B}$ contradicting the assumption that $\mathbb{B}$ contains exactly one 1-active ordering constraint. Thus, $\operatorname{supp}(\bar{x}) \cap Z^\star_{\tilde{\ell}} \neq \varnothing$, which proves that $\hat{\imath} \in Z^\star_{\tilde{\ell}}$. ◊

We are now able to prove Claim 27. Let $\bar{x}$ be a fractional vertex of P and let $\mathbb{B}$ be a reduced basis of $\bar{x}$. Our aim is to find a solution $\tilde{x} \neq \bar{x}$ of the system $A_{\mathbb{B}} x = b_{\mathbb{B}}$. To this end, we initialize $\tilde{x} = \bar{x}$ and proceed with the following steps. Let $\lambda > 0$ be a real parameter.

(1) Let $\check{\imath} \in [n]$ be the smallest index such that $\bar{x}_{\check{\imath}} > 0$ and $\check{\imath}$ is contained in the support of every ordering constraint in $\mathbb{B}$. Assign $\tilde{x}_{\check{\imath}} = \bar{x}_{\check{\imath}} - \lambda$.
(2) For every fractionally active ordering constraint ℓ in $\mathbb{B}$, assign $\tilde{x}_{\hat{z}_\ell} = \bar{x}_{\hat{z}_\ell} - \lambda$.
(3) If there exists an 1-active ordering constraint ℓ in $\mathbb{B}$, denote by $\hat{\imath}$ the greatest index in $[\check{z}_\ell - 1]$ with $\bar{x}_{\hat{\imath}} > 0$. Assign $\tilde{x}_{\hat{\imath}} = \bar{x}_{\hat{\imath}} + \lambda$.
(4) If some partitioning constraint for $\ell \in [q]$ is violated by $\tilde{x}$, let j be an index in Z_ℓ with $\bar{x}_j > 0$ such that j is not contained in the support of an ordering constraint in $\mathbb{B}$. Assign $\tilde{x}_j = \bar{x}_j + \lambda$.

Note that the index $\check{\imath}$ in Step 1 is well-defined, since otherwise, $\mathbb{B}$ would contain a trivial ordering constraint.

Claim 33. *After Step 2, $\tilde{x}$ fulfills all fractionally active ordering constraints in $\mathbb{B}$ with equality.*

Proof. Since $\check{\imath}$ is contained in the support of every ordering constraint ℓ in $\mathbb{B}$, Step 1 decreases the value $\sum_{i=1}^{\check{z}_\ell - 1} \bar{x}_i$ by λ. Since Step 2 decreases the value $\bar{x}_{\hat{z}_\ell}$ for fractionally active ordering constraints by λ as well, all fractionally active ordering constraints in $\mathbb{B}$ are fulfilled with equality after Step 2. ◊

Claim 34. *After Step 3, $\tilde{x}$ fulfills all ordering constraints in $\mathbb{B}$ with equality.*

Proof. Observe that $\hat{\imath} \neq \hat{z}_{\tilde{\ell}}$ for every fractionally active ordering constraint $\tilde{\ell}$: Let ℓ' be the index of the unique 1-active ordering constraint in $\mathbb{B}$. If there existed a cycle index $\tilde{\ell} \in [q]$ with $\hat{z}_{\tilde{\ell}} = \hat{\imath}$, every $j \in Z_{\tilde{\ell}}$ with $\bar{x}_j > 0$ would be contained in the support of the 1-active ordering constraint ℓ'. But then Claim 32 would imply $\hat{\imath} \in Z^\star_{\tilde{\ell}}$ contradicting $\hat{z}_{\tilde{\ell}} = \hat{\imath}$.

For this reason, $\hat{\imath}$ is not contained in the support of any fractionally active ordering constraint. Hence, Step 3 does not affect fractionally active ordering constraints in $\mathbb{B}$. Thus, the value of $\sum_{i=1}^{\check{z}_{\ell'}-1} \bar{x}_i$ which was decreased by λ in Step 1 is increased by λ in Step 3. Consequently, $\tilde{x}$ fulfills the unique 1-active ordering constraint ℓ' in $\mathbb{B}$ with equality. ◊

Next, observe that the index $\hat{\imath}$ in Step 3 and $\check{\imath}$ have to be different: For the sake of contradiction, assume $\hat{\imath} = \check{\imath}$. If ℓ' is the index of the unique 1-active ordering constraint, the only possible entry of $\bar{x}$ on $[\check{z}_{\ell'} - 1]$ is $\check{\imath}$. Since $\hat{\imath} = \check{\imath}$, we have $\bar{x}_{\check{\imath}} = 1$. Thus, the ordering constraint for ℓ' is trivial which is a contradiction to Claim 28.

Claim 35. *After Step 4, $\tilde{x}$ fulfills all partitioning constraints and it fulfills all ordering constraints in $\mathbb{B}$ with equality.*

Proof. We distinguish two cases. In the first case, $\mathbb{B}$ does not contain any 1-active ordering constraint. By Claim 29, there exists for every ordering constraint $\ell \in [q]$ an index $j \in Z_\ell$

with $\bar{x}_j > 0$ that is not contained in the support of an ordering constraint in $\mathbb{B}$. Hence, the index j in Step 4 is well-defined for each $\ell \in [q]$. By adding the violation of the partitioning constraint on cycle ℓ to $\tilde{x}_j$, we can achieve that the partitioning constraint holds without affecting entries of ordering constraints in $\mathbb{B}$.

In the second case, $\mathbb{B}$ contains exactly one 1-active ordering constraint by Claim 30. Denote with ℓ' the index of this ordering constraint. Observe that no entry of $\bar{x}$ on $Z_{\ell'}$ was modified by Steps 1–3. Hence, it suffices to establish that the partitioning constraints on the remaining cycles are fulfilled. If for each $\ell \in [q] \setminus \{\ell'\}$ there exists an index $j \in Z_\ell$ that is not contained in any ordering constraint in $\mathbb{B}$, we can use the same argumentation as in the first case. Otherwise, we have that there exists exactly one $\tilde{\ell} \in [q] \setminus \{\ell'\}$ that does not provide an index $j \in Z_{\tilde{\ell}}$ with $\bar{x}_j > 0$ that is not contained in the support of any ordering constraint in $\mathbb{B}$, see Claim 31.

By Claim 32, we know that the index $\hat{\imath}$ in Step 3 is contained in $Z^\star_{\tilde{\ell}}$. On the one hand, if $\check{\imath} \in Z_{\tilde{\ell}}$, the partitioning constraint for $\tilde{\ell}$ holds because we decreased $\bar{x}_{\check{\imath}}$ by λ in Step 1 and we increased $\bar{x}_{\hat{\imath}}$ by λ in Step 3. Moreover, $\check{\imath} \in Z_{\tilde{\ell}}$ implies that $\sum_{i=1}^{\check{z}_{\tilde{\ell}}-1} \bar{x}_i = 0$. Hence, the ordering constraint of $\tilde{\ell}$ cannot be contained in $\mathbb{B}$ by Claim 28. Thus, this constraint was not modified in Step 2. Consequently, the partitioning constraint on $Z_{\tilde{\ell}}$ holds by Step 1 and 3.

On the other hand, if $\check{\imath} \notin Z_{\tilde{\ell}}$, there exists $\ell'' \in [q] \setminus \{\ell', \tilde{\ell}\}$ such that $\check{\imath} \in Z^\star_{\ell''} \cap [\check{z}_{\tilde{\ell}} - 1]$. Thus, $\bar{x}_{\hat{z}_{\tilde{\ell}}} \geq \sum_{i=1}^{\check{z}_{\tilde{\ell}}-1} \bar{x}_i \geq \bar{x}_{\check{\imath}} > 0$. Since every positive entry of $\bar{x}$ on $Z_{\tilde{\ell}}$ is contained in the support of an ordering constraint in $\mathbb{B}$ and $\hat{z}_{\tilde{\ell}} \neq \hat{\imath} \in Z^\star_{\tilde{\ell}}$ by Claim 32, we conclude that the ordering constraint of $\tilde{\ell}$ is fractionally active because ℓ' is the only 1-active ordering constraint. Hence, Step 1 keeps entries of $\bar{x}$ on $Z_{\tilde{\ell}}$ invariant and Step 2 decreases $\bar{x}_{\hat{z}_{\tilde{\ell}}}$ by λ. Since we increased $\bar{x}_{\hat{\imath}}$ by λ in Step 3, and $\hat{\imath} \in Z^\star_{\tilde{\ell}}$, the partitioning constraint for $\tilde{\ell}$ holds. Thus, we have to ensure that the partitioning constraints hold for the remaining cycles. But since these cycles admit an index that is not contained in any active ordering constraint, see Claim 32, we can modify this entry analogously to the first case to fulfill the remaining partitioning constraints without affecting ordering constraints in $\mathbb{B}$. Consequently, the claim follows. ◊

With the above results, we are finally able to show that Claim 27 holds: Since we modified $\bar{x}$ in at least two entries to obtain $\tilde{x}$, $\tilde{x} \neq \bar{x}$. Furthermore, we modified $\bar{x}$ only in positive entries. Hence, if λ is sufficiently small, the same non-negativity constraints are active in $\bar{x}$ and $\tilde{x}$. For this reason, $\mathbb{B}$ is a reduced basis for $\bar{x}$ and $\tilde{x}$, which shows that the system $A_{\mathbb{B}}x = b_{\mathbb{B}}$ has two distinct solutions, contradicting the assumption that $A_{\mathbb{B}}$ is regular. Therefore, a reduced basis cannot contain any ordering constraint. By the initial argumentation, each vertex of P is integral. □

Proposition 17. *Let $\gamma \in \mathcal{S}_n$ be a monotone permutation with q cycles in its disjoint cycle decomposition. Then there exists a monotone permutation $\tilde{\gamma} \in \mathcal{S}_{n+q}$ such that $\mathrm{P}^{\leq}_{\gamma}$ is a linear projection of $\mathrm{P}^{=}_{\tilde{\gamma}}$.*

Proof. For each cycle ζ_ℓ, $\ell \in [q]$, of γ, we introduce a new element z_ℓ and extend the coordinate set $[n]$ to $\mathcal{I} \coloneqq [n] \cup \{z_\ell \,:\, \ell \in [q]\}$. On the extended coordinate set $\mathcal{I}$, we

$\gamma = (1,3,5,7)(2,4,6)(8,9)$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

$\tilde{\gamma} = (1,3,5,z_2,7)(2,4,z_1,6)(8,z_3,9)$ | 1 | 2 | 3 | 4 | 5 | 8 | z_1 | z_2 | z_3 | 6 | 7 | 9

Figure 3. Illustration of the total order constructed in the proof of Proposition 17. On the left-hand side, we present the permutation γ and the newly constructed permutation $\tilde{\gamma}$. On the right-hand side, the corresponding total orders w.r.t. $\leq$ and $\sqsubseteq$ are given.

introduce the total order $\sqsubseteq$ which is given by

$$i \sqsubseteq j \quad :\Leftrightarrow \quad \begin{cases} \text{if } i,j \in \mathcal{A}^\gamma \text{ and } i \leq j, \\ \text{if } i = z_{\ell_1}, j = z_{\ell_2} \text{ for some } \ell_1, \ell_2 \in [q] \text{ and } \hat{z}_{\ell_1} \leq \hat{z}_{\ell_2}, \\ \text{if } i,j \in \mathcal{D}^\gamma \text{ and } i \leq j, \\ \text{if } i \in \mathcal{A}^\gamma \text{ and } j \in \mathcal{I} \setminus [n], \\ \text{if } i \in \mathcal{I} \setminus [n] \text{ and } j \in \mathcal{D}^\gamma, \\ \text{if } i \in \mathcal{A}^\gamma \text{ and } j \in \mathcal{D}^\gamma, \end{cases}$$

see Figure 3 for an illustration.

Furthermore, we extend γ to a permutation $\tilde{\gamma} \in \mathcal{S}_\mathcal{I} \cong \mathcal{S}_{n+q}$ that is defined as

$$\tilde{\gamma}(i) = \begin{cases} \gamma(i), & \text{if } i \in Z_\ell \setminus \{\max Z_\ell^\star\} \text{ for some } \ell \in [q], \\ z_\ell, & \text{if } i = \max Z_\ell^\star \text{ for some } \ell \in [q], \\ \hat{z}_\ell, & \text{if } i = z_\ell \text{ for some } \ell \in [q], \end{cases}$$

that is, z_ℓ is inserted between the last and second last element of Z_ℓ w.r.t. the order $\leq$. The descent points of $\tilde{\gamma}$ w.r.t. the total order $\sqsubseteq$ are exactly the points in $\mathcal{D}^\gamma$ and the ascent points are given by $\mathcal{I} \setminus \mathcal{D}^\gamma$. Moreover, the cycles of $\tilde{\gamma}$ correspond to the cycles ζ_ℓ of γ extended by z_ℓ. Thus, $\tilde{\gamma}$ is monotone because each cycle of $\tilde{\gamma}$ contains exactly one descent point.

Consider the linear projection $\pi\colon \mathbb{R}^{n+q} \to \mathbb{R}^n$, $\pi(\tilde{x})_i = \tilde{x}_i$ for all $i \in [n]$. We claim that $\mathrm{P}_\gamma^\leq$ is the projection of $\mathrm{P}_{\tilde{\gamma}}^=$ by π:

Let $\tilde{x}$ be a vertex of $\mathrm{P}_{\tilde{\gamma}}^=$. Since $\tilde{x}$ fulfills the partitioning constraints on the cycles of $\tilde{\gamma}$, the projection $\pi(\tilde{x})$ fulfills the packing constraints on the cycles of γ, because the cycle supports of γ are subsets of the supports of the cycles of $\tilde{\gamma}$. To prove that $\pi(\tilde{x})$ is a vertex of $\mathrm{P}_\gamma^\leq$, it thus suffices to show that the property described in Lemma 11 holds for $\pi(\tilde{x})$.

If $\tilde{x}_j = 0$ for all $j \in \mathcal{D}^\gamma$, then $\pi(\tilde{x})_j = 0$ for all $j \in \mathcal{D}^\gamma$. Hence, $\pi(\tilde{x}) \in \mathrm{P}_\gamma^\leq$ by Lemma 11. Contrary, if there exists $j \in \mathcal{D}^\gamma$ such that $\tilde{x}_j = 1$, Lemma 11 implies that there exists $i \in \mathcal{I} \setminus \mathcal{D}^\gamma$, $i \sqsubseteq \tilde{\gamma}(j)$, with $\tilde{x}_i = 1$. Because $\tilde{\gamma}(j) = \gamma(j) \in \mathcal{A}^\gamma$, we have $i \sqsubseteq z_\ell$ and $i \neq z_\ell$ for every $\ell \in [q]$. Consequently, $i \in \mathcal{A}^\gamma$ and Lemma 11 implies $\pi(\tilde{x}) \in \mathrm{P}_\gamma^\leq$.

For the reverse direction, let $\bar{x}$ be a vertex of $\mathrm{P}_\gamma^\leq$. We define a vertex $\tilde{x}$ of $\mathrm{P}_{\tilde{\gamma}}^=$ by assigning $\tilde{x}_i = \bar{x}_i$ for all $i \in [n]$. Additionally, we set $\tilde{x}_{z_\ell} = 1$ if there is no 1-entry in $\bar{x}$ on Z_ℓ. By this definition, each cycle of $\tilde{\gamma}$ contains exactly one 1-entry in $\tilde{x}$. Furthermore, $\tilde{x}$ is feasible for $\mathrm{P}_{\tilde{\gamma}}^=$: If there exists a descent point $j \in \mathcal{I}$ with $\tilde{x}_j = 1$, then $\bar{x}_j = 1$ since γ and $\tilde{\gamma}$ have the same descent points. Because $\bar{x}$ is a vertex of $\mathrm{P}_\gamma^\leq$, there exists $i \in \mathcal{A}^\gamma$, $i < \gamma(j)$, with $\bar{x}_i = 1$. But due to the definition of the lifting, $\tilde{x}_i = 1$ as well. Since $i \sqsubseteq \tilde{\gamma}(j)$ and $i \neq \tilde{\gamma}(j)$, Lemma 11 yields that $\tilde{x}$ is feasible for $\mathrm{P}_{\tilde{\gamma}}^=$. □

Theorem 18. *Let $\gamma \in \mathcal{S}_n$ be a monotone permutation without fixed points. Then $\mathrm{P}_\gamma^\leq$ is completely described by (6), (7c), and the packing constraints $\sum_{i \in Z_\ell} x_i \leq 1$ for all $\ell \in [q]$.*

Proof. Let $\gamma \in \mathcal{S}_n$ be a monotone permutation with q cycles in its disjoint cycle decomposition. By Proposition 17, there exists a monotone permutation $\tilde{\gamma} \in \mathcal{S}_{n+q}$ such that $\mathrm{P}_\gamma^\leq$

is a linear projection of $\mathrm{P}^{=}_{\tilde{\gamma}}$. In particular, the construction in the proof of Proposition 17 shows that $\mathrm{P}^{\leq}_{\gamma}$ is the orthogonal projection of

$$
\begin{aligned}
-\sum_{i \in [\check{z}_\ell - 1] \setminus \hat{Z}} x_i + x_{\hat{z}_\ell} &\leq 0, && \ell \in [q],\\
\sum_{i \in Z_\ell \cup \{z_\ell\}} x_i &= 1, && \ell \in [q],\\
-x_i &\leq 0, && i \in [n] \cup \{z_\ell \,:\, \ell \in [q]\},
\end{aligned}
$$

onto $[n]$. In fact, the index set of the sum in the first family of inequalities is correct because of the reordering of $[n]$ by the order $\sqsubseteq$. In the following, we compute this projection, and thus a complete linear description.

By the construction of the total order $\sqsubseteq$ in the proof of Proposition 17, we have $\check{z}_{\ell'} \sqsubseteq z_\ell$, $\check{z}_{\ell'} \neq z_\ell$, for all ℓ, $\ell' \in [q]$. Hence, the only constraints in this system that involve variables z_ℓ are the partitioning constraints and non-negativity constraints. Solving the partitioning constraints for x_{z_ℓ}, $\ell \in [q]$, and substituting this term in the non-negativity constraint of x_{z_ℓ} eliminates the variables x_{z_ℓ} from the system and leads to the new inequalities $\sum_{i \in Z_\ell} x_i \leq 1$. Since this new system consists exactly of the Constraints (7c), the packing constraints for the cycles of γ, and the modified ordering constraints (6), the assertion follows. ☐

Theorem 19. *Let $\gamma \in \mathcal{S}_n$ be a monotone permutation without fixed points and assume that $\check{z}_1 < \check{z}_2 < \dots < \check{z}_q$. Then a facet of $\mathrm{P}^{\leq}_{\gamma}$ is defined by*

- *all packing constraints $\sum_{i \in Z_\ell} x_i \leq 1$, $\ell \in [q]$,*
- *(6) if and only if $\ell \neq 1$, and*
- *(7c) if and only if $i \in \{2, \dots, n\} \setminus \{\hat{z}_1\}$ or $i = 1$ and $\{2, \dots, \check{z}_2 - 1\} \setminus \{\hat{z}_1\} \neq \varnothing$.*

Proof. Throughout this proof we denote the characteristic vector of a set $S \subseteq [n]$ by the n-dimensional vector $\chi(S)$. Note that (6) implies for the cycle ζ_1 that $x_{\hat{z}_1} \leq 0$. Hence, this inequality and the non-negativity constraint $x_{\hat{z}_1} \geq 0$ show that $x_{\hat{z}_1} = 0$ holds for all $x \in \mathrm{P}^{\leq}_{\gamma}$. Hence, $\dim(\mathrm{P}^{\leq}_{\gamma}) \leq n-1$. In fact, $\dim(\mathrm{P}^{\leq}_{\gamma}) = n-1$, because the null vector, $\chi(i)$, $i \in [n] \setminus \hat{Z}$, as well as $\chi(\{1, \hat{z}_\ell\})$, $\ell \in \{2, \dots, q\}$, fulfill all inequalities of the complete linear description. Thus, these n affinely independent vectors are contained in $\mathrm{P}^{\leq}_{\gamma}$, proving $\dim(\mathrm{P}^{\leq}_{\gamma}) = n - 1$. In particular, this shows that the remaining inequalities cannot be implicit equations since neither of these inequalities reduces to $x_{\hat{z}_1} \geq 0$ or $x_{\hat{z}_1} \leq 0$.

Let $Ax \leq b$ be the system defined by (6), (7c), and packing constraints without $x_{\hat{z}_1} \geq 0$ and $x_{\hat{z}_1} \leq 0$. To investigate which of these inequalities define facets of $\mathrm{P}^{\leq}_{\gamma}$, we consider systems $A'x \leq b'$ which are obtained by removing one of the inequalities in $Ax \leq b$. Since $Ax \leq b$ together with $x_{\hat{z}_1} = 0$ is a complete linear description of $\mathrm{P}^{\leq}_{\gamma}$ in which no inequality is a positive multiple of another inequality or an implicit equation, the removed inequality is facet defining if and only if the feasible region of $A'x \leq b'$, $x_{\hat{z}_1} = 0$, is greater than the feasible region of $Ax \leq b$, $x_{\hat{z}_1} = 0$.

Observe that $\sum_{i \in Z_\ell} x_i \leq 1$ is the only inequality of $Ax \leq b$ that cuts off the vector $2\chi(\check{z}_\ell)$. Hence, the packing inequality defines a facet for every $\ell \in [q]$.

To prove that (6) defines a facet if $\ell \neq 1$, note that $x_{\hat{z}_\ell}$ only appears in its non-negativity constraint and (6) for parameter ℓ. Thus, if (6) is removed, the infeasible vector $\chi(\hat{z}_\ell)$ becomes feasible for $A'x \leq b'$. Hence, (6) defines a facet if $\ell \neq 1$. For $\ell = 1$, on the contrary, the ordering constraint cannot define a facet because it is an implicit equation.

Finally, we consider non-negativity inequalities $x_i \geq 0$. If $i \in \{2, \dots, n\} \setminus \{\hat{z}_1\}$, consider the vector $\chi(1) - \chi(i)$. This vector fulfills except for $x_i \geq 0$ all non-negativity constraints as well as all packing and ordering constraints. Thus, $x_i \geq 0$ defines a facet. To check whether $x_1 \geq 0$ defines a facet, observe that the ordering constraint for $\ell = 2$ is given by $x_{\hat{z}_2} \leq \sum_{i \in [\check{z}_2 - 1] \setminus \{\hat{z}_1\}} x_i$. Hence, non-negativity of $x_{\hat{z}_2}$ implies that $x_1 \geq 0$ cannot define a facet if $I \coloneqq \{2, \dots, \check{z}_2 - 1\} \setminus \{\hat{z}_1\} = \varnothing$. If $I \neq \varnothing$, the vector $\chi(j) - \chi(1)$, $j \in I$, fulfills all constraints except $x_1 \geq 0$. Hence, $x_1 \geq 0$ is facet defining. Thus, the only case in which $x_i \geq 0$, $i \in [n] \setminus \{\hat{z}_1\}$, does not define a facet is when $I = \varnothing$ and $i = 1$. This proves the assertion, since we have to exclude the implicit equation $x_{\hat{z}_1} \geq 0$. □

Proposition 23. *System (8) describes an integral polytope, and thus, together with the projection π it is an extended formulation of $\mathrm{P}^{\gtrsim}_{\gamma}$.*

Proof. To show integrality of the polytope induced by (8), we show that (8) is totally dual integral (TDI). To this end, we introduce dual variables μ, λ, κ, and ν which correspond to the Inequalities (8a)–(8d). Let $(w^x, w^y) \in \mathbb{Z}^{n \times q}$ be an objective for the System (8). Then the dual system is given by

$$\min \ \mu$$

$$\lambda_i - \nu_\ell \geq w^x_i, \qquad \ell \in [q],\ i \in Z^\star_\ell, \tag{15a}$$

$$\kappa_\ell - \nu_\ell \geq w^x_{\hat{z}_\ell}, \qquad \ell \in [q], \tag{15b}$$

$$\mu - \sum_{r=\ell}^{q} \sum_{i \in Z^\star_r} \lambda_i - \sum_{r=\ell+1}^{q} \kappa_r + \sum_{r=\ell}^{q} \nu_r \geq w^y_\ell, \qquad \ell \in [q], \tag{15c}$$

$$\mu, \lambda, \kappa, \nu \geq 0.$$

To prove that (8) is TDI, we have to show that (15) always has an integer optimal solution (if a solution exists). Let $\bar{X} \coloneqq (\bar{\mu}, \bar{\lambda}, \bar{\kappa}, \bar{\nu})$ be an optimal solution of the dual system. If all λ-, κ-, and ν-variables are integral, (15c) implies an integral lower bound on μ. Since this constraint and $\mu \geq 0$ are the only restrictions on μ, optimality of $\bar{X}$ implies that $\bar{\mu}$ is integral as well. Hence, it suffices to prove that the dual system always has an optimal solution with integral λ, κ, and ν.

If $\bar{X}$ is a solution in which all λ- and ν-variables are integral but some κ-variables are fractional, we can generate another solution $\tilde{X}$ with the same objective value by rounding down fractional κ-variables. This solution $\tilde{X}$ is, in fact, feasible for (15), since rounding down positive values $\bar{\kappa}_\ell$ cannot violate the non-negativity constraints and (15b) because $\bar{\kappa}_\ell$ is by assumption the only fractional variable involved in such constraints. Finally, $\tilde{X}$ cannot violate (15c) since decreasing some κ-variables and keeping the remaining variables invariant increases the left-hand side value of this constraint. For this reason, it is sufficient to construct an optimal solution $\bar{X}$ with integer values for λ- and ν-variables.

To prove the existence of such an optimal solution, let $\vartheta(x) \coloneqq x - \lfloor x \rfloor$ be the fractional part of a real number x, and let $L \coloneqq \{\ell \in [q] \,:\, \vartheta(\bar{\nu}_\ell) \leq \sum_{i \in Z^\star_\ell} \vartheta(\bar{\lambda}_i)\}$. We define a new solution $X' = (\mu', \lambda', \kappa', \nu')$ via $\mu' = \bar{\mu}$, $\lambda'_i = \lfloor \bar{\lambda}_i \rfloor$ as well as

$$\kappa'_\ell = \begin{cases} \lfloor \bar{\kappa}_\ell \rfloor, & \text{if } \ell \in L, \\ \bar{\kappa}_\ell + \lceil \bar{\nu}_\ell \rceil - \bar{\nu}_\ell, & \text{otherwise,} \end{cases} \quad \text{and} \quad \nu'_\ell = \begin{cases} \lfloor \bar{\nu}_\ell \rfloor, & \text{if } \ell \in L, \\ \lceil \bar{\nu}_\ell \rceil, & \text{otherwise,} \end{cases}$$

for all $\ell \in [q]$ and $i \in Z^\star_\ell$, and show that X' is feasible for (15). Consequently, since the objective value of $\bar{X}$ and X' are the same, the assertion follows by the above arguments for solutions with integral values for λ- and ν-variables.

Since all entries of $\bar{X}$ are non-negative, rounding an entry down cannot violate a non-negativity constraint. Moreover, the manipulations of entries of $\bar{X}$ that are different from rounding down cannot decrease their value. Hence, all non-negativity inequalities are fulfilled by X'.

To show feasibility of X' for (15a) and (15b), observe that

$$w^x_{\hat{z}_\ell} \leq \bar{\kappa}_\ell - \bar{\nu}_\ell \leq \bar{\kappa}_\ell - \lfloor \bar{\nu}_\ell \rfloor$$

holds for every $\ell \in [q]$. Since $w^x_{\hat{z}_\ell}$ is integral, we can round the right-hand side of this inequality down and obtain $w^x_{\hat{z}_\ell} \leq \lfloor \bar{\kappa}_\ell \rfloor - \lfloor \bar{\nu}_\ell \rfloor$. Thus, (15b) holds for all $\ell \in L$. Moreover, exactly the same arguments can be used to show that X' fulfills (15a) for every $\ell \in L$ and $i \in Z^\star_\ell$. To show that X' fulfills (15b) if $\ell \notin L$, we estimate

$$w^x_{\bar{z}_\ell} \leq \bar{\kappa}_\ell - \bar{\nu}_\ell = \bar{\kappa}_\ell + \lceil \bar{\nu}_\ell \rceil - \bar{\nu}_\ell - \lceil \bar{\nu}_\ell \rceil = \kappa'_\ell - \nu'_\ell.$$

Furthermore, if $\ell \notin L$, we have $\vartheta(\bar{\lambda}_i) < \vartheta(\bar{\nu}_\ell)$ for every $i \in Z^\star_\ell$. Hence,

$$\lambda'_i - \nu'_\ell = \lfloor \bar{\lambda}_i \rfloor - \lceil \bar{\nu}_\ell \rceil \geq w^x_i$$

holds, because $\bar{\lambda}_i - \bar{\nu}_\ell \geq w^x_i$ and

$$\begin{aligned}\mathbb{Z} \ni \lambda'_i - \nu'_\ell = \bar{\lambda}_i - \vartheta(\bar{\lambda}_i) - (\bar{\nu}_\ell + 1 - \vartheta(\bar{\nu}_\ell)) \;&\geq\; w^x_i + \vartheta(\bar{\nu}_\ell) - \vartheta(\bar{\lambda}_i) - 1 \\ &\overset{\ell \notin L}{>} w^x_i - 1 \in \mathbb{Z}.\end{aligned}$$

This shows that X' fulfills (15a) for all $\ell \notin L$ and $i \in Z^\star_\ell$

Thus, it remains to prove that X' does not violate (15c) for any $\ell \in [q]$. To this end, define $\xi = 1$ if $\ell \notin L$ and $\xi = 0$ otherwise. Then

$$\begin{aligned}&\sum_{r=\ell}^{q} \sum_{i \in Z^\star_r} (\bar{\lambda}_i - \lambda'_i) + \sum_{r=\ell+1}^{q} (\bar{\kappa}_r - \kappa'_r) - \sum_{r=\ell}^{q} (\bar{\nu}_r - \nu'_r) \\ &= \sum_{\substack{r=\ell \\ r \in L}}^{q} \Big(\sum_{i \in Z^\star_r} \vartheta(\bar{\lambda}_i) - \vartheta(\bar{\nu}_r) \Big) + \sum_{\substack{r=\ell+1 \\ r \in L}}^{q} \vartheta(\bar{\kappa}_r) + \sum_{\substack{r=\ell \\ r \notin L}}^{q} \sum_{i \in Z^\star_r} \vartheta(\bar{\lambda}_i) \\ &\quad + \sum_{\substack{r=\ell+1 \\ r \notin L}}^{q} (\bar{\kappa}_r - \kappa'_r - \bar{\nu}_r + \nu'_r) - \xi(\bar{\nu}_\ell - \lceil \bar{\nu}_\ell \rceil) \\ &\geq \sum_{\substack{r=\ell+1 \\ r \notin L}}^{q} (\bar{\kappa}_r - \bar{\kappa}_r - \lceil \bar{\nu}_r \rceil + \bar{\nu}_r - \bar{\nu}_r + \lceil \bar{\nu}_r \rceil) = 0.\end{aligned}$$

Consequently,

$$\begin{aligned}\mu' - \sum_{r=\ell}^{q} \sum_{i \in Z^\star_r} \lambda'_i - \sum_{r=\ell+1}^{q} \kappa'_r + \sum_{r=\ell}^{q} \nu'_r &= \bar{\mu} - \sum_{r=\ell}^{q} \sum_{i \in Z^\star_r} \bar{\lambda}_i + \sum_{r=\ell}^{q} \sum_{i \in Z^\star_r} (\bar{\lambda}_i - \lambda'_i) - \sum_{r=\ell+1}^{q} \bar{\kappa}_r \\ &\quad + \sum_{r=\ell+1}^{q} (\bar{\kappa}_r - \kappa'_r) + \sum_{r=\ell}^{q} \bar{\nu}_r - \sum_{r=\ell}^{q} (\bar{\nu}_r - \nu'_r) \\ &\geq \bar{\mu} - \sum_{r=\ell}^{q} \sum_{i \in Z^\star_r} \bar{\lambda}_i - \sum_{r=\ell+1}^{q} \bar{\kappa}_r + \sum_{r=\ell}^{q} \bar{\nu}_r \geq w^y_\ell,\end{aligned}$$

proving that X' fulfills (15c) and thus all constraints of (15). ☐

Lemma 24. *After eliminating variables $y_q, \dots, y_{k+1}$, $k \in [q-1]$, in System (9), the resulting system is given by (10).*

Proof. We prove this statement inductively. If $k = q$, both System (9) and (10) coincide. Thus, the induction base holds. For the inductive step, we assume that eliminating variables $y_q, \dots, y_{k+1}$ via FME from System (9) leads to System (10) with parameter k. Hence, it remains to show that we obtain System (10) for parameter $k-1$ if we eliminate y_k from System (10) for parameter k.

To this end, we have to combine Inequalities (10a) and (10b), which have a positive coefficient for y_k, with Inequalities (10c)–(10g), in which y_k has a negative coefficient. In the following, we compute these combinations, and we explain how the resulting inequality contributes to (10) for parameter $k-1$.

(10a)+(10c): $-\sum_{i \in Z_k} x_i \leq -1$. This inequality extends Family (10i).

(10a)+(10d): $-x_i \leq 0$ for all $i \in Z_k$. These inequalities extend Family (10h).

(10a)+(10e): $\sum_{r=1}^{k-1} y_r \leq 1$. This inequality is Inequality (10a) in System (10) for parameter $k-1$.

(10a)+(10f): $-\sum_{i \in Z_\ell} x_i + |Z_\ell^\star| \sum_{r=k+1}^{\ell} x_{\hat{z}_r} \leq b_\ell^{k-1} - 1$ for all $\ell \in [q] \setminus [k]$. Observe that these inequalities are dominated by the sum of (10i) and multiples of the upper bound constraints $x_{\hat{z}_r} \leq 1$, $r \in \{k, \dots, q\}$. Since both types of constraints are contained in System (10) for parameter $k-1$, the derived inequalities are redundant.

(10a)+(10g): $-x_i + \sum_{r=k+1}^{\ell} x_{\hat{z}_r} \leq \ell - k$ for all $i \in Z_k^\star$. By the same argument as in the previous case, these inequalities are redundant since $\sum_{r=k+1}^{\ell} x_{\hat{z}_r} \leq \ell - k$ by (10j) and $-x_i \leq 0$.

(10b)+(10c): $-|Z_k^\star| \sum_{r=1}^{k-1} y_r - \sum_{i \in Z_k} x_i + |Z_k^\star| x_{\hat{z}_k} \leq -1$. This inequality extends Family (10f).

(10b)+(10d): $-x_i + x_{\hat{z}_k} - \sum_{r=1}^{k-1} y_r \leq 0$ for all $i \in Z_k$. These inequalities extend Family (10g).

(10b)+(10e): $x_{\hat{z}_k} \leq 1$. This inequality extends Family (10j).

(10b)+(10f): $-|Z_\ell^\star| \sum_{r=1}^{k-1} y_r - \sum_{i \in Z_\ell} x_i + |Z_\ell^\star| \sum_{r=k}^{\ell} x_{\hat{z}_r} \leq b_\ell^{k-1} - 1$ for all indices $\ell \in \{k + 1, \dots, q\}$. These inequalities extend Family (10f).

(10b)+(10g): $-x_i + \sum_{r=k}^{\ell} x_{\hat{z}_r} - \sum_{r=1}^{k-1} y_r \leq \ell - k$ for all $\ell \in \{k+1, \dots, q\}$, $i \in Z_\ell^\star$. These inequalities extend Family (10g).

Together with the inequalities in System (10) for parameter k that are independent from k, the newly generated inequalities define System (10) for parameter $k-1$, proving the assertion. □

Appendix B. Detailed Results for Margot's Instances

In this section, we provide a detailed overview on the results obtained for the `Margot` test set. While Table 5 reports on the results obtained for the `default` setting, Tables 6–8 provide the test results for the settings `OF`, `symre`, and `pp-symre`, respectively.

Table 5. Detailed results for the `Margot` test set using setting `default`.

Setting	#nodes	time	#act	method-time	sym-time
cov954	37 744	99.7	0	0.0	0.0
cov1053	1 485 630	3600.0	0	0.0	0.0
cov1054	297 159	3600.0	0	0.0	0.0
cov1075	1 069 981	3600.0	0	0.0	0.0
cov1076	1 163 507	3600.0	0	0.0	0.0
cov1174	415 312	3600.0	0	0.0	0.0
sts27-complemented	2747	1.5	0	0.0	0.0
sts45-complemented	45 884	23.4	0	0.0	0.0
sts63-complemented	7 108 394	3471.4	0	0.0	0.0
sts81-complemented	5 118 836	3600.0	0	0.0	0.0
cod83	1 285 523	3600.0	0	0.0	0.0
cod83r	558 762	1115.1	0	0.0	0.0
cod93	154 939	3600.0	0	0.0	0.0
cod93r	285 288	3600.0	0	0.0	0.0
cod105	78 927	3600.0	0	0.0	0.0
cod105r	5101	263.6	0	0.0	0.0

Table 6. Detailed results for the `Margot` test set using setting `OF`.

Setting	#nodes	time	#act	method-time	sym-time
cov954	410	3.7	1	0.0	0.0
cov1053	298 662	738.1	1	1.3	0.0
cov1054	291 091	3600.0	1	1.0	0.0
cov1075	32 838	166.9	1	0.1	0.1
cov1076	1 172 953	3600.0	1	3.6	0.0
cov1174	543 693	3600.0	1	2.9	0.1
sts27-complemented	163	0.2	1	0.0	0.0
sts45-complemented	19 616	11.6	1	0.0	0.0
sts63-complemented	65 089	38.0	1	0.2	0.0
sts81-complemented	5 237 611	3600.0	1	12.3	0.0
cod83	21 856	57.9	1	0.1	0.0
cod83r	7503	21.5	1	0.0	0.0
cod93	252 617	3600.0	1	2.2	0.0
cod93r	576 015	3600.0	1	2.0	0.0
cod105	493	146.6	1	0.1	0.1
cod105r	5101	263.2	1	0.1	0.0

Table 7. Detailed results for the `Margot` test set using setting `symre`.

Setting	#nodes	time	#act	method-time	sym-time
cov954	914	5.5	1	0.0	0.0
cov1053	16 984	56.9	1	0.2	0.0
cov1054	4400	90.1	1	0.0	0.0
cov1075	2889	28.3	1	0.0	0.0
cov1076	19 085	113.5	1	0.1	0.0
cov1174	348 153	3600.0	1	3.4	0.1
sts27-complemented	2	0.1	1	0.0	0.0
sts45-complemented	2871	3.3	1	0.0	0.0
sts63-complemented	12 834	15.8	1	0.0	0.0
sts81-complemented	70 908	77.0	1	1.3	0.0
cod83	678	8.9	1	0.0	0.0
cod83r	241	5.9	1	0.0	0.0
cod93	34 440	323.5	1	0.6	0.0
cod93r	81 624	839.4	1	0.9	0.0
cod105	6	189.1	1	0.1	0.0
cod105r	2	53.2	1	0.0	0.1

Table 8. Detailed results for the `Margot` test set using setting `pp-symre`.

Setting	#nodes	time	#act	method-time	sym-time
cov954	914	5.6	1	0.0	0.0
cov1053	16 984	57.4	1	0.2	0.0
cov1054	4400	91.4	1	0.0	0.0
cov1075	2889	28.4	1	0.0	0.0
cov1076	19 085	112.4	1	0.1	0.0
cov1174	345 736	3600.0	1	3.1	0.1
sts27-complemented	2	0.1	1	0.0	0.0
sts45-complemented	2871	3.4	1	0.0	0.0
sts63-complemented	12 834	16.1	1	0.0	0.0
sts81-complemented	70 908	76.9	1	1.1	0.0
cod83	16	2.3	1	0.1	0.0
cod83r	375	5.3	1	0.0	0.0
cod93	19 066	174.6	1	0.5	0.0
cod93r	20 317	196.5	1	0.5	0.0
cod105	1	1.1	1	0.2	0.0
cod105r	1	0.5	1	0.1	0.1

References

[1] F. A. Aloul, K. A. Sakallah, and I. L. Markov. Efficient symmetry breaking for Boolean satisfiability. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), 2003.

[2] B. Assarf, E. Gawrilow, K. Herr, M. Joswig, B. Lorenz, A. Paffenholz, and T. Rehn. Computing convex hulls and counting integer points with `polymake`. *Mathematical Programming Computation*, 9(1):1–38, 2017.

[3] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1–3):3–44, 1998.

[4] E. Balas and R. Jeroslow. Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics*, 23(1):61–69, 1972.

[5] COLOR02 – computational symposium: Graph coloring and its generalizations. Available at http://mat.gsia.cmu.edu/COLOR02, 2002.

[6] M. Conforti, G. Cornuéjols, and G. Zambelli. Extended formulations in combinatorial optimization. *Annals of Operations Research*, 204(1):97–143, 2013.

[7] E. J. Friedman. Fundamental domains for integer programs with symmetries. In A. Dress, Y. Xu, and B. Zhu, editors, *Combinatorial Optimization and Applications*, volume 4616 of *LNCS*, pages 146–153. Springer Berlin Heidelberg, 2007.

[8] A. H. Gebremedhin, A. Tarafdar, F. Manne, and A. Pothen. New acyclic and star coloring algorithms with application to computing Hessians. *SIAM Journal on Scientific Computing*, 29(3):1042–1072, 2007.

[9] A. H. Gebremedhin, A. Tarafdar, A. Pothen, and A. Walther. Efficient computation of sparse Hessians using coloring and automatic differentiation. *INFORMS Journal on Computing*, 21(2):209–223, 2009.

[10] A. Ghoniem and H. D. Sherali. Defeating symmetry in combinatorial optimization via objective perturbations and hierarchical constraints. *IIE Transactions*, 43(8):575–588, 2011.

[11] A. Gleixner, L. Eifler, T. Gally, G. Gamrath, P. Gemander, R. L. Gottwald, G. Hendel, C. Hojny, T. Koch, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, F. Schlösser, F. Serrano, Y. Shinano, J. M. Viernickel, S. Vigerske, D. Weninger, J. T. Witt, and J. Witzig. The SCIP optimization suite 5.0. Technical report, ZIB, Takustr. 7, 14195 Berlin, 2017.

[12] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Springer, Berlin, 1988.

[13] C. Hojny and M. E. Pfetsch. A polyhedral investigation of star colorings. *Discrete Applied Mathematics*, 208:59–78, 2016.

[14] C. Hojny and M. E. Pfetsch. Polytopes associated with symmetry handling. *Mathematical Programming*, 175(1):197–240, 2019.

[15] IBM. IBM ILOG CPLEX Optimization Studio. https://www.ibm.com/products/ilog-cplex-optimization-studio, year = 2019.

[16] T. Januschowski and M. E. Pfetsch. Branch-cut-and-propagate for the maximum k-colorable subgraph problem with symmetry. In T. Achterberg and J. C. Beck, editors, *CPAIOR*, volume 6697 of *Lecture Notes in Computer Science*, pages 99–116. Springer, 2011.

[17] T. Januschowski and M. E. Pfetsch. The maximum -colorable subgraph problem and orbitopes. *Discrete Optimization*, 8(3):478–494, 2011.

[18] T. Junttila and P. Kaski. bliss: A tool for computing automorphism groups and canonical labelings of graphs. http://www.tcs.hut.fi/Software/bliss/, 2012.

[19] V. Kaibel and A. Loos. Branched polyhedral systems. In F. Eisenbrand and F. B. Shepherd, editors, *Integer Programming and Combinatorial Optimization, 14th International Conference, IPCO 2010, Lausanne, Switzerland, June 9–11, 2010. Proceedings*, volume 6080 of *Lecture Notes in Computer Science*, pages 177–190. Springer, 2010.

[20] V. Kaibel and M. E. Pfetsch. Packing and partitioning orbitopes. *Mathematical Programming*, 114(1):1–36, 2008.

[21] V. Kaibel, M. Peinhardt, and M. E. Pfetsch. Orbitopal fixing. *Discrete Optimization*, 8(4):595–610, 2011.

[22] G. Katsirelos, N. Narodytska, and T. Walsh. On the complexity and completeness of static constraints for breaking row and column symmetry. In: Principles and Practice of Constraint Programming – CP 2010, 2010.

[23] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems.* Springer, Berlin, 2004.

[24] D. Klabjan, G. L. Nemhauser, and C. Tovey. The complexity of cover inequality separation. *Operations Research Letters*, 23(1–2):35–40, 1998.

[25] T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, E. Danna, G. Gamrath, A. M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D. E. Steffy, and K. Wolter. Miplib 2010. *Mathematical Programming Computation*, 3(2):103–163, 2011.

[26] L. Liberti. Reformulations in mathematical programming: automatic symmetry detection and exploitation. *Mathematical Programming*, 131(1-2):273–304, 2012.
[27] A. Loos. *Describing Orbitopes by Linear Inequalities and Projection Based Tools*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2010.
[28] E. M. Luks and A. Roy The complexity of symmetry-breaking formulas. *Annals of Mathematics and Artificial Intelligence*, 41(1):19–45, 2004.
[29] F. Margot. Pruning by isomorphism in branch-and-cut. *Mathematical Programming*, 94(1):71–90, 2002.
[30] F. Margot. Exploiting orbits in symmetric ILP. *Mathematical Programming*, 98(1–3):3–21, 2003.
[31] F. Margot. Symmetry in integer linear programming. In M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, editors, *50 Years of Integer Programming*, pages 647–686. Springer, 2010.
[32] R. K. Martin, R. L. Rardin, and B. A. Campbell. Polyhedral characterization of discrete dynamic programming. *Operations Research*, 38(1):127–138, February 1990.
[33] B. D. McKay and A. Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60(0):94–112, 2014.
[34] MIPLIB 2017, http://miplib.zib.de, 2018.
[35] J. Ostrowski. *Symmetry in Integer Programming*. PhD thesis, Lehigh University, 2008.
[36] J. Ostrowski, M. F. Anjos, and A. Vannelli. Symmetry in scheduling problems, 2010.
[37] http://www.mathematik.tu-darmstadt.de/~pfetsch/symmetries.html.
[38] V. V. Vazirani. *Approximation Algorithms*. Springer, Berlin [i.a.], 2 edition, 2003.

Combinatorial Optimization Group, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Email address: c.hojny@tue.nl