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 [6] or Liberti [22]. If the symmetries form a group $\Gamma$ that reorders the columns of binary $(m \times n)$-matrices, Kaibel and Pfetsch [19] introduced the concept of full orbitopes to handle these symmetries. The full orbitope $O_{m,n}(\Gamma)$ is the convex hull of all binary matrices whose columns are sorted lexicographically maximal w.r.t. reorderings by permutations in $\Gamma$. 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 [19]. In the case of $\Gamma$ 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 [23]. Thus, the corresponding symmetry handling inequalities cannot be separated in polynomial time unless $P = NP$.

A different approach for deriving symmetry handling inequalities for arbitrary binary programs was discussed in [13]. There, we have introduced for a symmetry $\gamma$ the symresack $P_\gamma$, which is the convex hull of all binary vectors that are lexicographically not smaller than their images w.r.t. $\gamma$. Thus, 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 $P_\gamma$, we can completely handle the symmetries of $\gamma$. In [13], we derived such an IP formulation of exponential size that can be separated.

Date: May 2018.
in almost linear time, and we used this concept to derive an IP formulation of full orbitopes of linear size. However, complete linear descriptions of $P_\gamma$ are unknown in general.

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 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 a subset of the variable index set, $k$ is a positive integer, and $\sim \in \{\leq, =, \geq\}$. They cover the case of upper and lower bound constraints as well as equality constraints. After formally introducing cardinality constrained symresacks in Section 2, we develop an optimization algorithm for cardinality constrained symresacks that runs in cubic time in 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 thus 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 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 $\pm 1$. 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 outperform symmetry handling inequalities based on unconstrained symresacks in certain applications. 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 [13]. These properties will be useful in our investigation of cardinality constrained symresacks, which will be defined below.

A symmetry of a binary program

$$\max \{ w^\top x : Ax \leq b, x \in \{0, 1\}^n \},$$

(1)
where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $w \in \mathbb{R}^n$, is a permutation $\gamma$ of the set $[n] := \{1, \ldots, n\}$ with the following two properties: First, $\gamma(x) := (x_{\gamma^{-1}(1)}, \ldots, x_{\gamma^{-1}(n)})$ fulfills the constraints of (1) if and only if $x$ fulfills these constraints, i.e., $\gamma$ transforms feasible solutions to feasible solutions. Second, $\gamma$ 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 $S_n$, the group containing all permutations of $[n]$. Since computing the symmetry group of a binary program is NP-hard, see Margot [26], one often refrains from computing the whole symmetry group. Instead, one typically considers subgroups $\Gamma$ of the symmetry group that keep the problem formulation invariant, and only handles the symmetries contained in $\Gamma$. In the following, we always assume that $\Gamma$ 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.

To obtain a fundamental domain, Friedman [6] considered the fundamental domain inequalities (FD-inequalities)

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

and he showed that a vector $x \in \{0,1\}^n$ fulfills all inequalities of type (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. In particular, 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. $\Gamma$ from a binary program.

Note that Friedman’s approach neglects the group structure of the symmetry group $\Gamma$, 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, the FD-inequalities used by Friedman are impractical since their large coefficients might cause numerical problems. To overcome these numerical issues, we suggested in [13] to consider symresacks.

**Definition 1.** Let $\gamma \in S_n$. The symresack w.r.t. $\gamma$ is the polytope

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

that is, the vertices of $P_\gamma$ are the binary vectors $x$ that are 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 $P_\gamma$ can be used to handle the symmetries of a single permutation $\gamma$ 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 $\gamma$ can be obtained by deriving a facet description of $P_\gamma$. Moreover, any IP formulation of $P_\gamma$ has the same symmetry reducing effect as FD-inequalities. Thus, by finding an IP formulation of $P_\gamma$ with 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 $P_\gamma$ is a knapsack polytope. Note, however, that symresacks are not classical knapsack polytopes because (2) has positive and negative coefficients. However, $P_\gamma$ 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 [3], can be used to obtain an IP formulation of $P_\gamma$ 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, which is possible in $O(\alpha(n))$ time, where $\alpha: \mathbb{Z}^+ \to \mathbb{Z}^+$ is the inverse Ackermann function, see [13].

As a consequence, all symmetries of a binary program can be handled by replacing the FD-instabilities caused by FD-inequalities by separating minimal cover inequalities, which is possible in $O(|\Gamma|\alpha(n))$ time.

In the following, we extend the concept of symresacks by adding certain cardinality restrictions to symresacks. Analyzing these constrained symresacks, which are the main object of this paper, allows to find symmetry handling inequalities that take cardinality restrictions to symresacks. For the remainder of this paper, we assume that a permutation $\gamma \in S_n$ is given via its disjoint cycle decomposition $\gamma = \zeta_1 \ldots \zeta_q$. We denote the number of cycles in the disjoint cycle decomposition of $\gamma$ by $q = q(\gamma)$. The support of a cycle $\zeta_\ell$ is defined as $Z_\ell := \{i \in [n] : \zeta_\ell(i) \neq i\}$.

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

\[
P_\gamma^{\leq k} := \text{conv}\left(\left\{x \in P_\gamma \cap \{0,1\}^n : \sum_{i \in Z_\ell} x_i \leq k_\ell, \ell \in [q]\right\}\right),
\]

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

\[
P_\gamma^{\geq k} := \text{conv}\left(\left\{x \in P_\gamma \cap \{0,1\}^n : \sum_{i \in Z_\ell} x_i \geq k_\ell, \ell \in [q]\right\}\right),
\]

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 $P_\gamma^{\leq k}$ coincides with the ordinary symresack $P_\gamma$, and if $k$ is the null vector, then $P_\gamma^{\geq k} = P_\gamma$.

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 [13] to derive the optimization algorithm for symresacks. However, we refrain from providing the arguments for their validity. Instead, we refer the reader to [13].

**Property 3.** The vertex set of $P_\gamma$ 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 $(C_j^c)_{j}$ of $[n]$ such that
   - every set $C_j^c$ is a subset of a set $Z_\ell$ for some $\ell \in [q]$,
   - $V^c$ is empty if and only if $C_{j(c)}^c \cap C_{j(c) - 1}^c \neq \emptyset$, where $j(i)$ assigns every $i \in [n]$ the index $j(i)$ such that $i \in C_j^{c(j(i))}$, and
   - computing the partition $(C_j^c)_{j}$ and checking $C_{j(c)}^c \cap C_{j(c) - 1}^c \neq \emptyset$ can be done in $O(n)$ time.
over solution. Consequently, the theorem holds if we can show that the optimization problem
found by maximizing
of
P
between variables in different sets. Moreover, Property 3 implies that each set
the complexity bounds for the sets
in linear time. As a consequence, we can assume in the following that
Property 3,
time, respectively.

Theorem 4. Let \( \gamma \in S_n \) and \( k \in \mathbb{Z}_+^q \). The linear optimization problem over

- \( P_{\gamma}^{\leq k} \) can be solved in \( O(n^2 k) \) time,
- \( P_{\gamma}^{= k} \) can be solved in \( O(n^2 \min\{k, n-k\}) \) time, and
- \( P_{\gamma}^{\geq k} \) can be solved in \( O(n^2 (n-k)) \) time.

Proof. Let \((V^c)_{c \in [n+1]}\) be the partition of the vertices of \( P_\gamma \) mentioned in Property 3. Since
the constrained symresacks \( P_{\gamma}^{\leq k}, \sim \in \{\leq, =, \geq\} \), are subpolytopes of \( P_\gamma \) and all of their
vertices are binary, the family \( P_{\gamma}^{c,\sim k} := V^c \cap P_{\gamma}^{\sim k}, c \in [n+1] \), is a partition of the vertices
of \( P_{\gamma}^{\sim k} \). Hence, a maximizer of a linear objective \( w^\top x \), where \( w \in \mathbb{R}^n \), over \( P_{\gamma}^{\sim k} \) 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 \( P_{\gamma}^{c,\leq k}, P_{\gamma}^{c,= k}, \) and \( P_{\gamma}^{c,\geq k} \) can be solved in \( O(nk), O(n \min\{k, n-k\}) \), and \( O(n(n-k)) \)
time, respectively.

Let \( c \in [n+1] \) and let \( (C_j^c)_j \) be the partition of \( V^c \) described in Property 3. Due to
Property 3, \( (C_j^c)_j \) can be constructed and emptiness of \( V^c \) (and thus \( P_{\gamma}^{c,\sim k} \)) can be decided in
linear time. As a consequence, we can assume in the following that \( V^c \neq \emptyset \). To prove the
complexity bounds for the sets \( P_{\gamma}^{c,\sim k} \), we use the second part of Property 3, which
states that variables whose indices are contained in \( C_j^c \) are fixed to 1 and variables with
index in \( C_j^c(\gamma-1(c)) \) are fixed to 0. Furthermore, variables whose indices are contained in
the same set \( C_j^c \) have to obtain the same value from \( \{0, 1\} \), but there are no implications
between variables in different sets. Moreover, Property 3 implies that each set \( C_j^c \) is a
subset of a cycle \( Z_\ell \) of \( \gamma \) for some \( \ell \in [q(\gamma)] \).

Since there does not exist a dependence between variables of different sets \( C_j^c \) and each
such set is contained in exactly one cycle, the solutions on the different cycles of \( \gamma \) can
be computed independently. Furthermore, by introducing a variable \( y_C \) for every \( C \in C_\ell \),
where \( C_\ell \) denotes the sets \( C_j^c \) that are contained in \( Z_\ell \), we can model that variables in the
same set \( C \) have to obtain the same binary value. Thus, a maximal solution \( x \) in \( P_{\gamma}^{c,\sim k} \)
is given by setting \( x_i = y_{C_j^c(i)} \) for the maximal solution \( y \) of the subproblem that is defined
on the cycle that contains \( C_j^c(i) \). Consequently, the maximization problem over \( P_{\gamma}^{c,\sim k} \)
decomposes into the maximization problems
\[
\max \left\{ \sum_{C \in \mathcal{C}_\ell} w(C)y_C : \sum_{C \in \mathcal{C}_\ell} |C|y_C \sim k\ell, \ y_{C_{\gamma-1(\ell)}} = 1, \ y_{C_{\gamma-1(\ell)}} = 0 \right\}, \quad \text{if } C_{\ell} \in \mathcal{C}_\ell,
\]
\[
\max \left\{ \sum_{C \in \mathcal{C}_\ell} w(C)y_C : \sum_{C \in \mathcal{C}_\ell} |C|y_C \sim k\ell \right\}, \quad \text{otherwise},
\]
where \( \ell \in [q] \) and \( w(C) := \sum_{i \in C} w_i \).

For \( \mathcal{P}_{\gamma}^{\leq k} \), each of the above subproblems is a knapsack problem with knapsack inequality \( \sum_{C \in \mathcal{C}_\ell} |C|y_C \leq k\ell \). Since \( |\mathcal{C}_\ell| \leq |Z_\ell| \), 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. [20]. Consequently, the knapsack subproblems on the different cycles can be solved in \( \mathcal{O}(\sum_{\ell \in [q]} |Z_\ell|k\ell) \subseteq \mathcal{O}(n\hat{k}) \) time, because the cycle supports \( Z_\ell \) form a partition of \([n]\).

For \( \mathcal{P}_{\gamma}^{\geq k} \), let \( a^\top y := \sum_{C \in \mathcal{C}_\ell} |C|y_C \geq k\ell \) 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:
\[
\max \left\{ \sum_{C \in \mathcal{C}_\ell} w(C)y_C : a^\top y \geq k\ell, \ y \in \{0, 1\}^{\mid Z_\ell} \right\}
\]
\[
= \max \left\{ \sum_{C \in \mathcal{C}_\ell} w(C) - \sum_{C \in \mathcal{C}_\ell} w(C)\tilde{y}_C : a^\top(\mathbb{1} - \tilde{y}) \geq k\ell, \ \tilde{y} \in \{0, 1\}^{\mid Z_\ell} \right\}
\]
\[
= \max \left\{ \sum_{C \in \mathcal{C}_\ell} w(C) - \sum_{C \in \mathcal{C}_\ell} w(C)\tilde{y}_C : a^\top \tilde{y} \leq a^\top \mathbb{1} - k\ell, \ \tilde{y} \in \{0, 1\}^{\mid Z_\ell} \right\}.
\]
Since \( a^\top \mathbb{1} = |Z_\ell| \), the subproblem on cycle \( \mathcal{C}_\ell \) can be solved via dynamic programming in time \( \mathcal{O}(|Z_\ell|(n\hat{k} - k\ell)) \). For this reason, all subproblems can be solved in
\[
\mathcal{O}(\sum_{\ell \in [q]} |Z_\ell|(|Z_\ell| - k\ell)) \subseteq \mathcal{O}(n^2 - n\hat{k})
\]
which concludes the proof.

Due to the equivalence of optimization and separation, Theorem 4 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 \( \pi : \mathbb{R}^d \to \mathbb{R}^n \) 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 5.** Let \( \gamma \in \mathcal{S}_n \) and \( k \in \mathbb{Z}_+^d \). The extension complexity of
\begin{itemize}
  \item \( \mathcal{P}_{\gamma}^{\leq k} \) is in \( \mathcal{O}(n^2\hat{k}) \),
  \item \( \mathcal{P}_{\gamma}^{= k} \) is in \( \mathcal{O}(n^2 \min\{\hat{k}, n - \hat{k}\}) \), and
  \item \( \mathcal{P}_{\gamma}^{\geq k} \) is in \( \mathcal{O}(n^2(n - \hat{k})) \).
\end{itemize}

**Proof.** Given a collection \( (P^c)_c \) of polytopes, the concept of disjunctive programming allows to define an extended formulation of \( P := \text{conv}(\bigcup_c P^c) \) by combining complete linear descriptions, see Balas [2], or extended formulations of \( P^c \); see Kaibel and Loos [18]. 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
\[
P_{\gamma}^{\leq k} = \text{conv} \left( \bigcup_{c=1}^{n+1} \text{conv}(P_{\gamma}^{c,\leq k}) \right),
\]
we can use disjunctive programming to find an extended formulation of \( P_{\gamma}^{\leq k} \) provided that we know extended formulations of \( \text{conv}(P_{\gamma}^{c,\leq k}) \). To find these extended formulations, we use a classical result of Martin et al. [27].

Given a combinatorial optimization problem that can be solved by a dynamic programming algorithm, Martin et al. [27] 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. Thus, since the optimization problem over \( P_{\gamma}^{c,\leq k} \) can be solved by a dynamic programming approach, see the proof of Theorem 4, it suffices to construct the corresponding hypergraph to obtain an extended formulation of \( \text{conv}(P_{\gamma}^{c,\leq k}) \).

Recall that the optimization problem over \( P_{\gamma}^{c,\leq k} \) decomposes into \( q(\gamma) \) many knapsack problems (3). Due to Conforti et al. [5], there exists a hypergraph \( G_\ell \) corresponding to the feasible region of the \( \ell \)-th knapsack problem \( a^\top x \leq k_\ell, \ x \in \{0,1\}^n \), in the Decomposition (3) whose size is in \( \mathcal{O}(\vert Z_\ell \vert k) \) and which meets the requirements of Martin et al. Because the Subproblems (3) can be solved independently, we can connect the hypergraphs \( G_\ell, \ \ell \in \{q\} \), in series to model the optimization problem over \( P_{\gamma}^{c,\leq k} \) (and thus over \( \text{conv}(P_{\gamma}^{c,\leq k}) \)) as a flow problem in a hypergraph \( G \). Because \( G \) consists of \( \mathcal{O}(\sum_{\ell \in \{q\}} |Z_\ell| k) = \mathcal{O}(n^k) =: \mathcal{O}(s^c) \) nodes and arcs, the LP model of Martin et al. is of size \( \mathcal{O}(s^c) \) as well, see Conforti et al. [5, Theorem 7.1]. Thus, there exists an extended formulation of \( P_{\gamma}^{\leq k} \) of size \( \mathcal{O}(\sum_{\ell \in \{q\}} n^k) = \mathcal{O}(n^2 k) \) by the initial arguments in this proof.

To show the assertion for \( P_{\gamma}^{\leq k} \) and \( P_{\gamma}^{\geq k} \), 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 \( \min \{k, n-k\} \) and \( n-k \), respectively, see the proof of Theorem 4. Hence, since the size of the hypergraph constructed above only depends on the size of the cycles \( \zeta_\ell \) of \( \gamma \) 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 \( P_{\gamma}^{\leq k} \) and \( P_{\gamma}^{\geq k} \), which concludes the proof. \( \square \)

Remark 6. For the all-ones vector \( k = 1 \), Theorems 4 and 5 yield that the complexity of maximizing a linear objective over \( P_{\gamma}^{\geq k} \) as well as the extension complexity of \( P_{\gamma}^{\geq k} \) are contained in \( \mathcal{O}(n^3) \). In fact, one can show that the linear optimization over \( P_{\gamma}^{\geq k} \) is solvable in \( \mathcal{O}(n^2) \) time and that \( k \)-covering symresacks admit an extended formulation of \( \mathcal{O}(n^3) \) size.

Remark 7. Theorems 4 and 5 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 \( \mathcal{O}(n^3) \).

Since constrained symresacks admit a compact extended formulation, we are theoretically able to derive complete linear descriptions of \( P_{\gamma}^{\leq k}, P_{\gamma}^{\geq k}, \) and \( P_{\gamma}^{\geq k} \) by computing the projection of the corresponding extended formulations. However, computing this projection is quite complicated. Hence, complete linear descriptions are 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 \( k = 1 \), 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 \( \mathbb{I} \)-packing, \( \mathbb{I} \)-partitioning, and \( \mathbb{I} \)-covering symresacks. The corresponding polytopes are denoted by \( P_{\leq \gamma} \), \( P_{= \gamma} \), and \( P_{\geq \gamma} \), respectively. Moreover, observe that if \( \gamma \) contains a fixed point \( i \), then \( \gamma \) 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 \( \gamma \) 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 \( \gamma \in S_n \), we investigate packing and partitioning symresacks in this section. First, we derive a linear size integer programming formulation of \( P_{\leq \gamma} \) and \( P_{= \gamma} \) with ternary coefficients. Afterwards, we prove that this IP formulation already defines a complete linear description for packing and partitioning symresacks of particular permutations. Before we start our analysis, we present an example that illustrates the application of packing and partitioning symresacks.

Example 8. Let \( G = (V, E) \) be an undirected graph and let \( k \) be a positive integer. A \( k \)-coloring of \( G \) is an assignment of the \( k \) colors to the nodes of \( G \) such that adjacent nodes are colored differently. A \( k \)-star coloring of \( G \) is a \( k \)-coloring such that each path subgraph of \( G \) containing four nodes is colored by at least three different colors. The maximum \( k \)-star colorable subgraph problem (\( M_k \text{SCSC} \)) is to find a node maximum induced subgraph of \( G \) that is \( k \)-star colorable. A natural IP formulation of this problem is

\[
\begin{align*}
\text{max} & \quad \sum_{v \in V} \sum_{j=1}^{k} x_{vj} \\
\text{subject to} & \quad \sum_{j=1}^{k} x_{vj} \leq 1, \quad v \in V, \quad (4a) \\
& \quad x_{uj} + x_{vj} \leq 1, \quad \{u, v\} \in E, \ j \in [k], \quad (4b) \\
& \quad \sum_{v \in P} (x_{vi} + x_{vj}) \leq 3, \quad P \in \mathcal{P}, \ i, j \in [k], \ i \neq j, \quad (4c) \\
\end{align*}
\]

where \( x_{vj} = 1 \) if and only if node \( v \) is colored with color \( j \) and \( \mathcal{P} \) denotes the set of all path subgraphs of \( G \) containing four nodes. Star colorings arise, for example, in the efficient computation of sparse Hessian matrices, see, e.g., Gebremedhin et al. [7, 8].

\( M_k \text{SCS} \) consists of two kinds of symmetries. First, we can associate with every relabeling \( \gamma \in S_k \) of the colors the permutation \( \gamma' \in S_{V \times [k]} \) that maps entry \((v, j)\) to \((v, \gamma(j))\). Thus, each cycle of \( \gamma' \) 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 \( \gamma' \).

Second, each automorphism \( \gamma: V \to V \) of \( G \) gives rise to a permutation \( \gamma' \) of the rows of \( x \). Thus, each cycle of \( \gamma' \) is a subset of a column of \( x \). In particular, if \( \gamma \) 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 \( \gamma' \) 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 $P_\gamma$, an IP formulation of constrained symresacks $P^\leq_\gamma$ and $P^=_\gamma$ 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 $P_\gamma$ 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 $P^\leq_\gamma$ and $P^=_\gamma$ 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 $\gamma$ by $A = A^\gamma := \{ i \in [n] : \gamma(i) > i \}$ and the set of all descent points of $\gamma$ by $D = D^\gamma := \{ i \in [n] : \gamma(i) < i \}$. Moreover, since we assume that $\gamma$ does not contain a fixed point, we have $A \cup D = [n]$.

**Lemma 9.** Let $\gamma \in S_n$ and let $\bar{x} \in \{0,1\}^n$ fulfill packing and partitioning constraints, respectively, on the cycles of $\gamma$. Then $\bar{x}$ is contained in $P^\leq_\gamma$ and $P^=_\gamma$, respectively, if and only if

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

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

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

The characterization of the vertices of $P^\leq_\gamma$ and $P^=_\gamma$ of Lemma 9 can easily be enforced by the inequalities

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

Furthermore, since all entries of a vertex $x$ of $P^\leq_\gamma$ and $P^=_\gamma$ 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 $P^\leq_\gamma$ and $P^=_\gamma$ stated in the following proposition.

**Proposition 10.** For every $\gamma \in S_n$, an IP formulation of $P^\leq_\gamma$ is given by packing constraints, non-negativity constraints, and (5). Moreover, by substituting packing constraints by partitioning constraint, we obtain an IP formulation for $P^=_\gamma$.

Proposition 10 shows that symmetries related to packing and partitioning symresacks can be handled by linearly many inequalities with ternary coefficients. In contrast to
Example 11. 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 \(\gamma \in S_n\) and let \(\gamma' \in S_{[m] \times [n]}\) be the permutation that reorders the columns of \(X\) according to \(\gamma\), cf. the color symmetries in Example 8. Then, a column reordering \(\gamma'\) is monotone if and only if the underlying permutation \(\gamma\) is monotone.

To analyze the IP formulations of \(P^{\leq}_\gamma\) and \(P^=\gamma\) for monotone permutations, we use the notation \(\hat{\ell} := \min\{i \in [\ell]\}\) and \(\hat{z}_\ell := \max\{i \in [\ell]\}\). Moreover, define \(\hat{Z}_\ell := [\ell] \setminus \{\hat{z}_\ell\}\).

Using this notation, Inequalities (5) simplify for monotone permutations to
\[- \sum_{i \in [\hat{z}_\ell - 1]} x_i + x_{\hat{z}_\ell} \leq 0, \quad \ell \in [q(\gamma)], \tag{6}\]
where \(\hat{Z} := \{\hat{z}_\ell : \ell \in [q]\}\). For partitioning symresacks, we will see below that the following inequalities already define a complete linear description:
\[- \sum_{i=1}^{\hat{z}_\ell - 1} x_i + x_{\hat{z}_\ell} \leq 0, \quad \ell \in [q(\gamma)], \tag{7a}\]
\[\sum_{i \in \hat{Z}_\ell} x_i = 1, \quad \ell \in [q(\gamma)], \tag{7b}\]
\[-x_i \leq 0, \quad i \in [n]. \tag{7c}\]

Observe that (7a) is slightly weaker than (6). However, the weaker inequalities suffice to define a complete linear description, since if there exists \([\hat{z}_\ell - 1] \cap \hat{Z} \neq \emptyset\), 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 12 illustrates the structure of ordering constraints, which might be helpful in the proof of Theorem 13 below.

Example 12. 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
\[x_{10} \leq 0, \quad -x_1 + x_{11} \leq 0, \quad -x_1 - x_2 - x_3 + x_9 \leq 0, \text{ and } -x_1 - x_2 - x_3 - x_4 + x_{12} \leq 0.\]
In particular, if \(N_\ell\) denotes the indices of variables that have a negative sign in the ordering constraint of cycle \(\ell\), then \(N_\ell \subseteq N_{\hat{\ell}}\) if and only if \(\hat{z}_\ell \leq \hat{z}_{\hat{\ell}}\).

Theorem 13. Let \(\gamma \in S_n\) be a monotone permutation without fixed points. Then \(P^=\gamma\) is completely described by (7).

Proof. Let \(P = \text{conv}\{x \in \mathbb{R}^n : x \text{ fulfills (7a)--(7c)}\}\) and abbreviate the defining inequality/equation system of \(P\) with \(Ax \leq b\). Note that \(P^=\gamma = \text{conv}(P \cap \mathbb{Z}^n)\) by Proposition 10.
Hence, if we can establish that each vertex of \(P\) is integral, we have proven that \(P = P^=\gamma\).
A basis \( B \) 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 \( A_B \) such that \( \bar{x} \) is the unique solution of \( A_B \bar{x} = b_B \), where the subscript \( B \) denotes the restriction of \( A \) and \( b \) to the constraints in \( B \). 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 [19] to show validity of a complete linear description of partitioning orbitopes.

Given a vertex \( \bar{x} \) of \( P \), we call a basis \( B \) 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 of \( \bar{x} \) 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 \( B \) which does not contain any ordering constraint. Consequently, \( B \) contains \( q \) partitioning constraints and \( n - q \) non-negativity constraints. The latter implies that \( \bar{x} \) has exactly \( q \) non-zero entries. But since \( \bar{x} \) fulfills all partitioning constraints, each cycle support \( Z_{\ell} \) 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 14.** 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 \( Z_{\ell} \) 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 \( A_B \bar{x} = b_B \) if \( B \) 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 \( B \) be a reduced basis that contains an ordering constraint for \( \ell \in [q] \). We call this constraint trivial if either \( \sum_{i=1}^{\bar{z}_\ell-1} \bar{x}_i = \bar{x}_{\bar{z}_\ell} = 0 \) or \( \bar{x}_{\bar{z}_\ell} = 1 \) and there exists exactly one \( j \in [\bar{z}_\ell - 1] \) such that \( \bar{x}_j = 1 \). In the latter case, \( \bar{x}_i = 0 \) holds for all \( i \in [\bar{z}_\ell - 1] \setminus \{ j \} \) since the ordering constraint is contained in \( B \).

**Claim 15.** A reduced basis \( B \) 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 \( B \) of \( \bar{x} \). Denote with \( \ell \in [q] \) the index of this trivial ordering constraint.

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

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

$$
\sum_{i \in Z_{t_1}} x_i - \sum_{i \in Z_{t_2}} x_i + \sum_{i \in [z_{t_2} - 1] \setminus \{j\}} (-x_i) + \sum_{i \in Z_{t_2}^*} (-x_i) - \sum_{i \in Z_{t_1} \setminus \{j\}} (-x_i) = -z_{t_2} - 1 - \sum_{i=1} x_i + x_{z_{t_2}}.
$$

Next, we prove some structural properties of the distribution of non-zero entries of $\bar{x}$ on each support $Z_{t_1}$. These properties will be crucial to prove Claim 14. Before we proceed with the first property described in Claim 16, 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 $\bar{x}_{z_{t_2}} = 1$, we call the ordering constraint 1-active. Otherwise, if the ordering constraint is fulfilled with equality but $\bar{x}_{z_{t_2}} < 1$, it is called fractionally active.

**Claim 16.** Let $\mathcal{B}$ 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 $j \in Z_{t_1}$ that is not contained in the support of an ordering constraint in $\mathcal{B}$ such that $\bar{x}_j > 0$.

*Proof.* If $\bar{x}_{z_{t_2}} = 1$ for any $\ell \in [q]$, the ordering constraint for $\ell$ cannot be contained in $\mathcal{B}$ since $\mathcal{B}$ does not contain any 1-active ordering constraint. Moreover, $\bar{x}_{z_{t_2}}$ cannot be contained in any ordering constraint in $\mathcal{B}$, because this would imply that this other ordering constraint would be 1-active as well. Thus, there exists $j \in Z_{t_1}$, namely $j = z_{t_2}$, such that $\bar{x}_j > 0$ and $j$ is not contained in the support of any ordering constraint in $\mathcal{B}$. Consequently, it suffices to consider only those cycles $\ell' \in [q]$ for which $\bar{x}_{z_{t_2}} < 1$.

If the claim was false, there would be $\ell' \in [q]$ with $\bar{x}_{z_{t_2}} < 1$ such that each $j \in Z_{t_2}$ with $\bar{x}_j > 0$ is contained in the support of a fractionally active ordering constraint. Denote with $j$ the greatest index in $Z_{t_2}$ with $\bar{x}_j > 0$, and let $\ell$ be the index of a cycle with $j < z_{t_2}$ and whose ordering constraint is contained in $\mathcal{B}$. The indices $j$ and $\ell$ are well-defined because $\sum_{j \in Z_{t_2}} \bar{x}_j = 1$ and $\bar{x}_{z_{t_2}} < 1$ as well as all positive variables in $Z_{t_2}$ are contained in the support of an active ordering constraint.

By definition of $j$ and $\ell$ as well as by the partitioning constraints, we have

$$
\bar{x}_{z_{t_2}} + \sum_{i \in [z_{t_2} - 1] \cap Z_{t_2}^*} \bar{x}_i = 1
$$

and $\hat{z}_{t_2} \leq j < \hat{z}_{t_2}$.

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

$$
1 \geq \bar{x}_{z_{t_2}} = \sum_{i=1}^{\hat{z}_{t_2} - 1} \bar{x}_i = \sum_{i=1}^{\hat{z}_{t_2} - 1} \bar{x}_i + \sum_{i=\hat{z}_{t_2}}^{\hat{z}_{t_2} - 1} \bar{x}_i \geq \sum_{i=1}^{\hat{z}_{t_2} - 1} \bar{x}_i + \sum_{i \in [z_{t_2} - 1] \cap Z_{t_2}^*} \bar{x}_i
$$

$$
(8) \geq \sum_{i=1}^{\hat{z}_{t_2} - 1} \bar{x}_i + 1 - \bar{x}_{z_{t_2}} \geq \bar{x}_{z_{t_2}} + 1 - \bar{x}_{z_{t_2}} = 1.
$$

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

Consequently, for each $\ell \in [q]$ there exists an index $j \in Z_{t_1}$ with $\bar{x}_j > 0$ such that $j$ is not contained in the support of any ordering constraint in $\mathcal{B}$.

**Claim 17.** A reduced basis contains at most one 1-active ordering constraint.

*Proof.* Assume there exists a reduced basis $\mathcal{B}$ that contains two 1-active ordering constraints $\ell_1$ and $\ell_2$, where $\hat{z}_1 < \hat{z}_2$. Then $\bar{x}_{\hat{z}_1} = \bar{x}_{\hat{z}_2} = 1$ as well as $\bar{x}_i = 0$ for all $i \in Z_{\ell_1} \cup Z_{\ell_2} =: I_1 \cup I_2$. Moreover, 1-activeness of the ordering constraint for $\ell_1$
and $\tilde{z}_{\ell_1} < \tilde{z}_{\ell_2}$ implies that $\bar{x}_i = 0$ for all $i \in I_3 := \{\tilde{z}_{\ell_1}, \ldots, \tilde{z}_{\ell_2} - 1\}$, since otherwise, the ordering constraint for $\ell_2$ cannot be fulfilled with equality. Thus, $\bar{x}_i = 0$ for all $i \in I := I_1 \cup I_2 \cup I_3$.

Because $B$ is a reduced basis of $\bar{x}$, $B$ contains the non-negativity constraints for all indices in $I$. Thus, we can generate the left-hand side of the ordering constraint of $\ell_2$ by the following linear combination of non-negativity constraints for variables $\bar{x}_i$, $i \in I$, the ordering constraint for $\ell_1$ and the partitioning constraints for $\ell_1$ and $\ell_2$:

$$
\left( - \sum_{i=1}^{\tilde{z}_{\ell_1}-1} x_i + x_{\tilde{z}_{\ell_1}} \right) - \sum_{i \in Z_{\ell_1}} x_i - \sum_{i \in Z_{\ell_1}^*} (-x_i) + \sum_{i \in I_3} (-x_i) - \sum_{i \in Z_{\ell_2}} (-x_i) + \sum_{i \in Z_{\ell_2}^*} (-x_i) = - \sum_{i=1}^{\tilde{z}_{\ell_2}-1} x_i + x_{\tilde{z}_{\ell_2}}.
$$

Since $A_B$ has to be regular, this is a contradiction. □

**Claim 18.** Let $B$ be a reduced basis of $\bar{x}$ that contains exactly one 1-active ordering constraint $\ell' \in [q]$. Then there is at most one $\ell \in [q]$, $\ell \neq \ell'$, such that each $j \in Z_\ell$ with $\bar{x}_j > 0$ is contained in the support of an ordering constraint in $B$.

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

$$
\sum_{i \in Z_{\ell'} \cap \{z_{\ell'} - 1\}} \bar{x}_i = 1,
$$

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

$$
\sum_{i \in Z_\ell \cap \{z_{\ell'} - 1\}} \bar{x}_i = 0 \quad (9)
$$

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

On the one hand, assume $\bar{x}_{\tilde{z}_{\ell}} < 1$. Then there exists $j \in Z_\ell \setminus \{z_{\ell'} - 1\}$ with $\bar{x}_j > 0$ due to the partitioning constraint for $\ell$. Since $j$ is not contained in the support of the ordering constraint of $\ell'$, the index $j$ cannot be contained in the support of any ordering constraint in $B$. On the other hand, consider $\bar{x}_{\tilde{z}_{\ell}} = 1$. If $\tilde{z}_{\ell}$ is contained in the support of an ordering constraint in $B$, then this ordering constraint has to be 1-active. But since the only 1-active ordering constraint in $B$ is the ordering constraint of $\ell'$ and $\tilde{z}_{\ell} > z_{\ell'} - 1$ by (9), $\tilde{z}_{\ell}$ cannot be contained in the support of the ordering constraint of $\ell'$. Consequently, $\tilde{z}_{\ell}$ is not contained in the support of any ordering constraint in $B$. For this reason, $Z_\ell$ contains in both cases an element that is not contained in the support of an ordering constraint in $B$, which proves that there is at most one $\hat{\ell} \in [q] \setminus \{\ell'\}$ such that each $j \in Z_\ell$ with $\bar{x}_j > 0$ is contained in the support of an ordering constraint in $B$.

As a consequence, we can assume in the following that

$$
\sum_{i \in Z_\ell \cap \{z_{\ell'} - 1\}} \bar{x}_i < 1
$$

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

$$
\bar{x}_{\tilde{z}_{\ell}} + \sum_{i \in Z_{\hat{\ell}} \cap \{z_{\hat{\ell}} - 1\}} \bar{x}_i = 1 : \quad (10)
$$
Assume there exists \( j \in \mathbb{Z}_\ell^* \setminus [\tilde{z}_{\ell'} - 1] \) such that \( \bar{x}_j > 0 \). Since every positive entry of \( \bar{x} \) on \( \mathbb{Z}_\ell \) is contained in the support of an ordering constraint in \( B \), there exists \( \ell'' \in [q] \) whose ordering constraint is contained in \( B \) and that fulfills \( \tilde{z}_{\ell''} < \tilde{z}_{\ell'} \). But then the ordering constraint for \( \ell'' \) cannot be contained in \( B \) because

\[
\sum_{i=1}^{\tilde{z}_{\ell''}-1} \bar{x}_i \geq \sum_{i=1}^{\tilde{z}_{\ell'}-1} \bar{x}_i + \sum_{i=1}^{\tilde{z}_{\ell'}-1} \bar{x}_i = 1,
\]

a contradiction. Consequently, the only positive entry of \( \bar{x} \) on \( \mathbb{Z}_\ell \setminus [\tilde{z}_{\ell'} - 1] \) can be \( \tilde{z}_{\ell'} \).

In particular, since \( \tilde{z}_{\ell} \) has to be contained in the support of an ordering constraint in \( B \) and \( \sum_{i \in \mathbb{Z}_\ell \cap [\tilde{z}_{\ell'} - 1]} \bar{x}_i < 1 \) holds, this implies that the ordering constraint for \( \ell \) is contained in \( B \).

If \( \bar{x}_i > 1 \), the ordering constraint for \( \ell \) cannot be contained in \( B \) by assumption. Thus, we can assume that \( \bar{x}_i \in (0, 1) \), and, in particular, (10) then implies \( \tilde{z}_{\ell} < \tilde{z}_{\ell'} \). Since these arguments hold for each \( \ell \in [q] \) that fulfills (10), assume for the sake of contradiction that there exist \( \tilde{1}, \tilde{2} \in [q] \), \( \tilde{z}_{\tilde{1}} < \tilde{z}_{\tilde{2}} < \tilde{z}_{\ell'} \), such that both fulfill (10). With these assumptions, we have

\[
1 = \sum_{i=1}^{\tilde{z}_{\ell'} - 1} \bar{x}_i \geq \sum_{i=1}^{\tilde{z}_{\tilde{1}} - 1} \bar{x}_i + \sum_{i \in \mathbb{Z}_\ell^* \cap [\tilde{z}_{\ell'} - 1]} \bar{x}_i \geq \sum_{i \in \mathbb{Z}_\ell^* \cap [\tilde{z}_{\ell'} - 1]} \bar{x}_i + \sum_{i \in \mathbb{Z}_\ell^* \cap [\tilde{z}_{\ell'} - 1]} \bar{x}_i = \bar{x}_{\tilde{z}_{\tilde{1}}} + 1 - \bar{x}_{\tilde{z}_{\tilde{1}}} + 1 - \bar{x}_{\tilde{z}_{\tilde{2}}} = 2 - \bar{x}_{\tilde{z}_{\tilde{2}}} > 1.
\]

Because this inequality chain cannot be valid, there exists at most one \( \tilde{\ell} \in [q] \) which fulfills (10). Thus, the claim holds.

**Claim 19.** Let \( B \) be a reduced basis of \( \bar{x} \) that contains exactly one 1-active ordering constraint \( \ell' \in [q] \) and let \( i \) be the greatest index in \([\tilde{z}_{\ell'} - 1]\) with \( \bar{x}_i > 0 \). If there exists \( \tilde{\ell} \in [q] \), \( \tilde{\ell} \neq \ell' \), such that each \( j \in \mathbb{Z}_{\tilde{\ell}} \) with \( \bar{x}_j > 0 \) is contained in the support of an ordering constraint in \( B \), then \( i \in \mathbb{Z}_{\tilde{\ell}}^* \).

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

\[
1 = \bar{x}_{\tilde{z}_{\ell'}} = \sum_{i=1}^{\tilde{z}_{\ell'} - 1} \bar{x}_i \geq \sum_{i=1}^{\tilde{z}_{\tilde{\ell}} - 1} \bar{x}_i \geq \sum_{i \in \mathbb{Z}_{\tilde{\ell}}^*} \bar{x}_i = 1.
\]

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

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

\[
\bar{x}_{\tilde{z}_{\tilde{\ell}}} + \sum_{i \in \mathbb{Z}_{\tilde{\ell}} \cap [\tilde{z}_{\ell'} - 1]} \bar{x}_i = 1.
\]
This implies that each $j \in Z^*_\ell$ with $\bar{x}_j > 0$ is contained in the support of the ordering constraint for $\ell'$. Then,

$$1 = \bar{x}_{\ell'} - 1 = \sum_{i=1}^{\ell'-1} \bar{x}_i \geq \sum_{i=1}^{\ell'-1} \bar{x}_i + \sum_{i \in Z_{\ell'[\ell'-1]}^\ell} \bar{x}_i \geq \bar{x}_{\ell'} + \sum_{i \in Z_{\ell'[\ell'-1]}^\ell} \bar{x}_i = 1.$$

For this reason, $\bar{x}_j = 0$ for each $j \in \{\bar{x}_{\ell'}, \ldots, \bar{x}_{\ell' - 1}\} \setminus Z_\ell$. Moreover, $\bar{x}_{\ell'} < 1$, because otherwise, the ordering constraint for $\ell$ has to be 1-active and it has to be contained in $B$ contradicting the assumption that $B$ contains exactly one 1-active ordering constraint. Thus, $\text{supp}(\bar{x}) \cap Z^*_\ell \neq \emptyset$, which proves that $\bar{x} \in Z^*_\ell$.

We are now able to prove Claim 14. Let $\bar{x}$ be a fractional vertex of $P$ and let $B$ be a reduced basis of $\bar{x}$. Our aim is to find a solution $\bar{x} \neq \bar{x}$ of the system $A_Bx = b_B$. To this end, we initialize $\bar{x} = \bar{x}$ and proceed with the following steps. Let $\lambda > 0$ be a real parameter.

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

Note that the index $i$ in Step 1 is well-defined, since otherwise, $B$ would contain a trivial ordering constraint.

Claim 20. After Step 2, $\bar{x}$ fulfills all fractionally active ordering constraints in $B$ with equality.

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

Claim 21. After Step 3, $\bar{x}$ fulfills all ordering constraints in $B$ with equality.

Proof. Observe that $i \neq \hat{z}_i$ for every fractionally active ordering constraint $\hat{\ell}$. Let $\ell'$ be the index of the unique 1-active ordering constraint in $B$. If there existed a cycle index $\hat{\ell} \in [q]$ with $\hat{z}_\hat{\ell} = i$, every $j \in Z_{\hat{\ell}}$ with $\bar{x}_j > 0$ would be contained in the support of the 1-active ordering constraint $\ell'$. But then Claim 19 would imply $\hat{x} \not\in Z^*_\ell$ contradicting $\hat{z}_\hat{x} = i$.

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

Next, observe that the index $i$ in Step 3 and $i$ have to be different: For the sake of contradiction, assume $i = i$. If $\ell'$ is the index of the unique 1-active ordering constraint, the only possible entry of $\bar{x}$ on $[\hat{z}_{\ell'} - 1]$ is $i$. Since $\hat{i} = i$, we have $\bar{x}_i = 1$. Thus, the ordering constraint for $\ell'$ is trivial which is a contradiction to Claim 15.

Claim 22. After Step 4, $\bar{x}$ fulfills all partitioning constraints and it fulfills all ordering constraints in $B$ with equality.

Proof. We distinguish two cases. In the first case, $B$ does not contain any 1-active ordering constraint. By Claim 16, 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 $B$. 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 $\ell$ to $\bar{x}_j$, we can achieve that the partitioning constraint holds without affecting entries of ordering constraints in $B$.

In the second case, $B$ contains exactly one 1-active ordering constraint by Claim 17. Denote with $\ell'$ 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 $B$, 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 $B$, see Claim 18.

By Claim 19, we know that the index $\tilde{i}$ in Step 3 is contained in $Z_{\tilde{\ell}}^\ast$. On the one hand, if $\tilde{i} \in Z_{\tilde{\ell}}$, the partitioning constraint for $\tilde{\ell}$ holds because we decreased $\bar{x}_\tilde{i}$ by $\lambda$ in Step 1 and we increased $\bar{x}_\tilde{i}$ by $\lambda$ in Step 3. Moreover, $\tilde{i} \in Z_{\tilde{\ell}}$ implies that $\sum_{i=1}^{\tilde{\ell}-1} \bar{x}_i = 0$. Hence, the ordering constraint of $\tilde{\ell}$ cannot be contained in $B$ by Claim 15. 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 $\tilde{i} \notin Z_{\tilde{\ell}}$, there exists $\ell'' \in [q] \setminus \{\ell', \tilde{\ell}\}$ such that $\tilde{i} \in Z_{\ell''} \cap [\tilde{\ell} - 1]$. Thus, $\bar{x}_{\tilde{\ell}} \geq \sum_{i=1}^{\tilde{\ell}-1} \bar{x}_i \geq \bar{x}_{\tilde{i}} > 0$. Since every positive entry of $\bar{x}$ on $Z_{\tilde{\ell}}$ is contained in the support of an ordering constraint in $B$ and $\tilde{\ell} \neq \tilde{i} \in Z_{\ell}^\ast$ by Claim 19, we conclude that the ordering constraint of $\tilde{\ell}$ is fractionally active because $\ell'$ is the only 1-active ordering constraint. Hence, Step 1 keeps entries of $\bar{x}$ on $Z_{\tilde{\ell}}$ invariant and Step 2 decreases $\bar{x}_{\tilde{\ell}}$ by $\lambda$. Since we increased $\bar{x}_\tilde{i}$ by $\lambda$ in Step 3, and $\tilde{i} \in Z_{\tilde{\ell}}^\ast$, 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 19, we can modify this entry analogously to the first case to fulfill the remaining partitioning constraints without affecting ordering constraints in $B$. Consequently, the claim follows.

With the above results, we are finally able to show that Claim 14 holds: Since we modified $\bar{x}$ in at least two entries to obtain $\bar{x}, \bar{x} \neq \bar{x}$. Furthermore, we modified $\bar{x}$ only in positive entries. Hence, if $\lambda$ is sufficiently small, the same non-negativity constraints are active in $\bar{x}$ and $\bar{x}$. For this reason, $B$ is a reduced bases for $\bar{x}$ and $\bar{x}$, which shows that the system $A_B x = b_B$ has two distinct solutions, contradicting the assumption that $A_B$ is regular. Therefore, a reduced basis cannot contain any ordering constraint. By the initial argumentation, each vertex of $P$ is integral.

Remark 23. In general, Theorem 13 is false if $\gamma$ is not monotone. For example, consider the permutation $\gamma = (1, 7, 5, 3)(2, 8, 6, 4)$. Experiments with the software tool POLYMAKE [1] 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 $P_{\gamma}^\ast$.

By a projection argument, we are also able to prove that the IP formulation of packing symresacks provided in Proposition 10 already is a complete linear description of $P_{\gamma}^\ast$ if $\gamma$ is monotone.

Proposition 24. Let $\gamma \in S_n$ be a monotone permutation with $q$ cycles in its disjoint cycle decomposition. Then there exists a monotone permutation $\hat{\gamma} \in S_{n+q}$ such that $P_{\hat{\gamma}}^\ast$ is a linear projection of $P_{\gamma}^\ast$.

Proof. For each cycle $\zeta_\ell, \ell \in [q]$, of $\gamma$, we introduce a new element $z_\ell$ and extend the coordinate set $[n]$ to $\mathcal{I} := [n] \cup \{z_\ell : \ell \in [q]\}$. On the extended coordinate set $\mathcal{I}$, we
\[ \gamma = (1, 3, 5, 7)(2, 4, 6)(8, 9) \]

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

**Figure 1.** Illustration of the total order constructed in the proof of Proposition 24. On the left-hand side, we present the permutation \( \gamma \) and the newly constructed permutation \( \tilde{\gamma} \). On the right-hand side, the corresponding total orders w.r.t. the order \( \leq \) and \( \sqsubseteq \) are given.

introduce the total order \( \sqsubseteq \) which is given by

\[
i \sqsubseteq j \iff \begin{cases} 
  \text{if } i, j \in A^\gamma \text{ and } i \leq j, \\
  \text{if } i = z_{i_1}, j = z_{i_2} \text{ for some } \ell_1, \ell_2 \in [q] \text{ and } \hat{z}_{\ell_1} \leq \hat{z}_{\ell_2}, \\
  \text{if } i, j \in D^\gamma \text{ and } i \leq j, \\
  \text{if } i \in A^\gamma \text{ and } j \in I \setminus [n], \\
  \text{if } i \in I \setminus [n] \text{ and } j \in D^\gamma, \\
  \text{if } i \in A^\gamma \text{ and } j \in D^\gamma,
\end{cases}
\]

see Figure 1 for an illustration.

Furthermore, we extend \( \gamma \) to a permutation \( \tilde{\gamma} \in S_I \cong S_{n+q} \) that is defined as

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

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

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

Let \( \tilde{x} \) be a vertex of \( P^\leq \tilde{\gamma} \). 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 \( \gamma \), because the cycle supports of \( \gamma \) are subsets of the supports of the cycles of \( \tilde{\gamma} \). To prove that \( \pi(\tilde{x}) \) is a vertex of \( P^\leq \gamma \), it thus suffices to show that the property described in Lemma 9 holds for \( \pi(\tilde{x}) \).

If \( \tilde{x}_j = 0 \) for all \( j \in D^\gamma \), then \( \pi(\tilde{x})_j = 0 \) for all \( j \in D^\gamma \). Hence, \( \pi(\tilde{x}) \in P^\leq \gamma \) by Lemma 9. Contrary, if there exists \( j \in D^\gamma \) such that \( \tilde{x}_j = 1 \), Lemma 9 implies that there exists \( i \in I \setminus D^\gamma, i \sqsubseteq \tilde{\gamma}(j), \) with \( \tilde{x}_i = 1 \). Because \( \tilde{\gamma}(j) = \gamma(j) \in A^\gamma \), we have \( i \sqsubseteq z_\ell \) for every \( \ell \in [q] \). Consequently, \( i \in A^\gamma \) and Lemma 9 implies \( \pi(\tilde{x}) \in P^\leq \gamma \).

For the reverse direction, let \( \bar{x} \) be a vertex of \( P^\leq \gamma \). We define a vertex \( \tilde{x} \in P^\leq \tilde{\gamma} \) by assigning \( \bar{x}_i = \tilde{x}_i \) for all \( i \in [n] \). Additionally, we set \( \bar{x}_{z_\ell} = 1 \) if there is no 1-entry in \( \tilde{x} \) on \( Z_\ell \). By this definition, each cycle of \( \tilde{\gamma} \) contains exactly one 1-entry in \( \tilde{x} \). Furthermore, \( \tilde{x} \) is feasible for \( P^\leq \tilde{\gamma} \): If there exists a descent point \( j \in I \) with \( \tilde{x}_j = 1 \), then \( \bar{x}_j = 1 \) since \( \gamma \) and \( \tilde{\gamma} \) have the same descent points. Because \( \bar{x} \) is a vertex of \( P^\leq \gamma \), \( \tilde{x} \) is a vertex of \( P^\leq \tilde{\gamma} \), \( i \sqsubseteq \tilde{\gamma}(j), \) 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 9 yields that \( \bar{x} \) is feasible for \( P^\leq \gamma \).

**Theorem 25.** Let \( \gamma \in S_n \) be a monotone permutation without fixed points. Then \( P^\leq \gamma \) 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.** Let \( \gamma \in S_n \) be a monotone permutation with \( q \) cycles in its disjoint cycle decomposition. By Proposition 24, there exists a monotone permutation \( \tilde{\gamma} \in S_{n+q} \) such that \( P^\leq \gamma \)
is a linear projection of $P_{\gamma}^\ell$. In particular, the construction in the proof of Proposition 24 shows that $P_{\gamma}^\ell$ is the orthogonal projection of
\[ - \sum_{i \in [\ell-1] \setminus \hat{Z}} x_i + x_{\hat{z}} \leq 0, \quad \ell \in [q], \]
\[ \sum_{i \in Z_\ell \cup \{z_\ell\}} x_i = 1, \quad \ell \in [q], \]
\[-x_i \leq 0, \quad i \in [n] \cup \{z_\ell : \ell \in [q]\},\]
ono 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 $\subseteq$. In the following, we compute this projection, and thus a complete linear description.

By the construction of the total order $\subseteq$ in the proof of Proposition 24, we have $\hat{z}_\ell' \subseteq z_\ell$, $\hat{z}_\ell' \neq z_\ell$, for all $\ell, \ell' \in [q]$. Hence, the only constraints in this system that involve variables $z_\ell$ are the partitioning constraints and non-negativity constraints. Solving the partitioning constraints for $x_{\hat{z}_\ell}$, $\ell \in [q]$, and substituting this term in the non-negativity constraint of $x_{\hat{z}_\ell}$ eliminates the variables $x_{\hat{z}_\ell}$ 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 $\gamma$, and the modified ordering constraints (6), the assertion follows. □

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

**Theorem 26.** Let $\gamma \in S_n$ be a monotone permutation without fixed points and assume that $\hat{z}_1 < \hat{z}_2 < \cdots < \hat{z}_q$. Then a facet of $P_{\gamma}^\ell$ 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 $i \in \{2, \ldots, n\} \setminus \{\hat{z}_1\}$ or $i = 1$ and $\{2, \ldots, \hat{z}_2 - 1\} \setminus \{\hat{z}_1\} \neq \emptyset$.

**Proof.** Note that (6) implies for the cycle $\zeta_1$ 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 $x \in P_{\gamma}^\ell$. Hence, $\dim(P_{\gamma}^\ell) \leq n - 1$. In fact, $\dim(P_{\gamma}^\ell) = n - 1$, because the null vector, $\chi(i)$, $i \in [n] \setminus \hat{Z}$, as well as $\chi(\{1, \hat{z}_1\})$, $\ell \in \{2, \ldots, q\}$, fulfill all inequalities of the complete linear description. Thus, these $n$ affinely independent vectors are contained in $P_{\gamma}^\ell$, proving $\dim(P_{\gamma}^\ell) = n - 1$.

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 $P_{\gamma}^\ell$, 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 $P_{\gamma}^\ell$ 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 $2\chi(\hat{z}_\ell)$. 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 $\ell$. 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 $i \in \{2, \ldots, n\} \setminus \{\hat{z}_1\}$, consider the vector $\chi(1) - \chi(i)$. This vector fulfills besides $x_i \geq 0$ 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 $x_{z_2} \leq \sum_{i \in \{z_2-1\}\setminus\{z_1\}} x_i$. Hence, non-negativity of $x_{z_2}$ implies that $x_1 \geq 0$ cannot define a facet if $I := \{2, \ldots, z_2 - 1\} \setminus \{z_1\} = \emptyset$. If $I \neq \emptyset$, the vector $\chi(j) - \chi(1)$, $j \in I$, fulfills all constraints but $x_1 \geq 0$. Hence, $x_1 \geq 0$ is facet defining. Thus, the only case in which $x_i \geq 0$, $i \in [n] \setminus \{z_1\}$, does not define a facet is when $I = \emptyset$ and $i = 1$. This proves the assertion, since we have to exclude the implicit equation $x_{z_1} \geq 0$.

Due to Theorems 13 and 25, 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 $P^P_γ$ and $P^P_γ$. 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-inequality 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.

Finally, we remark that ordering constraints were already discussed in the literature for particular permutations. For example, Ostrowski et al. [30] used the ordering constraints (5) to handle symmetries related to permutations $γ$ that are compositions of consecutive 2-cycles, i.e., $γ = (1, 2)(3, 4)\ldots(2n - 1, 2n)$ for some positive integer $n$. 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 monotone permutations.

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 $P^C_γ$ for particular permutations.

**Example 27.** Let $C$ be a collection of subsets of $[n]$ such that $\bigcup_{C \in C} C = [n]$. The set covering problem is to find a smallest subcollection $C'$ of $C$ such that $\bigcup_{C \in C'} C = [n]$. A classical IP formulation of this problem is

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

where $C_i := \{C \in C : i \in C\}$, see Vazirani [32]. If there exists a symmetry $γ \in S_C$ of the IP formulation such that each support $Z$ of a cycle of $γ$ is a superset of a set $C_i$, then

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

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

Computer experiments with POLYMAKE [1] indicate that the facial structure of $P^C_γ$ 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 $\hat{z}_{\ell+1} = \hat{z}_\ell + 1$ for all $\ell \in [g(γ) - 1]$, i.e., the supports of permutation cycles are consecutive sets. The basis of our investigation is provided by the following lemma.
Lemma 28. Let $\tilde{x} \in \{0,1\}^n$ and let $\gamma \in S_n$ be a monotone and ordered permutation. Then $\tilde{x} \succeq \gamma(\tilde{x})$ if and only if

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

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

Observation 29. Let $\tilde{x} \in P^\pi_{\gamma}$ and let $\gamma \in S_n$ be a monotone and ordered permutation. If cycle $\zeta_\ell$ is a constant cycle of $\gamma$ w.r.t. $\tilde{x}$, then $\tilde{x}_i = 1$ for all $i \in \mathbb{Z}_\ell$ due to the covering constraints of $P^\pi_{\gamma}$.

In the following, we develop an extended formulation to derive a complete linear description of $P^\pi_{\gamma}$ for monotone and ordered permutations $\gamma \in S_n$. Due to Lemma 28 and Observation 29, a vertex $x$ of $P^\pi_{\gamma}$ is completely characterized by the index of the first non-constant cycle $\ell'$ and the entries of $x$ on $\mathbb{Z}^*_{\ell'}$ and $\bigcup_{\ell=\ell'+1}^{\ell'} \mathbb{Z}_\ell$. Hence, we can completely characterize $x$ by introducing a binary indicator $y_{\ell'}$, $\ell \in [q]$, that is 1 if and only if $\zeta_\ell$ 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 $\{\tilde{x}_{\ell'} \} \cup \bigcup_{\ell=1}^{\ell'-1} \mathbb{Z}_\ell$ by 0 and coincides with $x$ otherwise. In particular, we can generate $x$ from the pair $(\tilde{x}, y)$ via the map

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

where $\ell$ is the index of the cycle of $\gamma$ that contains $i$. Note that the definition of $\pi$, 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 $i \in \mathbb{Z}_\ell$ with $\ell < \ell'$, and $\sum_{i \in \mathbb{Z}_\ell} \tilde{x}_i \geq 1$ as well as $\sum_{r=1}^{\ell} y_r = 1$ for all remaining cycles $\ell$.

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

$$
\begin{align*}
\sum_{r=1}^{m} y_r & \leq 1, \quad \text{(11a)} \\
\tilde{x}_i & \leq \sum_{r=1}^{\ell} y_r, \quad \ell \in [q], \ i \in \mathbb{Z}^*_\ell, \quad \text{(11b)} \\
\tilde{x}_{\ell} & \leq \sum_{r=1}^{\ell-1} y_r, \quad \ell \in [q], \quad \text{(11c)} \\
\sum_{r=1}^{\ell} y_r & \leq \sum_{i \in \mathbb{Z}_\ell} \tilde{x}_i, \quad \ell \in [q], \quad \text{(11d)} \\
\tilde{x}_i & \geq 0, \quad i \in [n], \quad \text{(11e)} \\
y_{\ell'} & \geq 0, \quad \ell \in [q]. \quad \text{(11f)}
\end{align*}
$$
Inequality (11a) ensures that \( y \) declares at most one cycle as the first non-constant cycle. Inequalities (11b) and (11c) guarantee that \( \tilde{x}_i = 0 \) if \( i \) is contained in a cycle that appears before the first non-constant cycle. Additionally, (11c) enforces that \( \tilde{x}_{2i} = 0 \) if the index of the first non-constant cycle is greater than or equal to \( \ell \). Finally, Inequality (11d) 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 (11e) and (11f) ensure non-negativity of all variables. Hence, (11) is an IP formulation of the vectors \((\tilde{x}, y)\) that encode a vertex of \( \mathbb{P}^\gamma_{\gamma} \) via \( \pi \).

**Proposition 30.** System (11) describes an integral polytope, and thus, together with the projection \( \pi \) it is an extended formulation of \( \mathbb{P}^\gamma_{\gamma} \).

**Proof.** To show integrality of the polytope induced by (11), we show that (11) is totally dual integral (TDI). To this end, we introduce dual variables \( \mu, \lambda, \kappa, \) and \( \nu \) which correspond to the Inequalities (11a)–(11d). Let \((w^x, w^y) \in \mathbb{Z}^{n \times q}\) be an objective for the System (11). Then the dual system is given by

\[
\begin{align*}
\min \mu \\
\lambda_i - \nu_i \geq w_i^x, & \quad \ell \in [q], i \in Z_q^\ast, \quad (12a) \\
\kappa_\ell - \nu_\ell \geq w_{2\ell}^x, & \quad \ell \in [q], \quad (12b) \\
\mu - q \sum_{\ell=\ell}^{q-1} \lambda_i - \sum_{\ell=\ell+1}^q \kappa_\ell + q \sum_{\ell=\ell}^q \nu_\ell \geq w_\ell^y, & \quad \ell \in [q], \quad (12c) \\
\mu, \lambda, \kappa, \nu \geq 0.
\end{align*}
\]

To prove that (11) is TDI, we have to show that (12) always has an integer optimal solution (if a solution exists). Let \( \bar{X} := (\bar{\mu}, \bar{\lambda}, \bar{\kappa}, \bar{\nu}) \) be an optimal solution of the dual system. If all \( \lambda, \kappa, \) and \( \nu \)-variables are integral, (12c) implies an integral lower bound on \( \mu \). Since this constraint and \( \mu \geq 0 \) are the only restrictions on \( \mu \), 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 \( \lambda, \kappa, \) and \( \nu \).

Moreover, if \( \bar{X} \) is a solution in which all \( \lambda \)- and \( \nu \)-variables are integral but some \( \kappa \)-variables are fractional, we can generate another solution \( \tilde{X} \) with the same objective value by rounding down fractional \( \kappa \)-variables. This solution \( \tilde{X} \) is, in fact, feasible for (12), since rounding down positive values \( \bar{\kappa}_\ell \) cannot violate the non-negativity constraints and (12b) because \( \bar{\kappa}_\ell \) is by assumption the only fractional variable involved in such constraints. Finally, \( \tilde{X} \) cannot violate (12c) since decreasing some \( \kappa \)-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 \( \lambda \)- and \( \nu \)-variables.

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

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

for all \( \ell \in [q] \) and \( i \in Z_q^\ast \), and show that \( X' \) is feasible for (12). 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 \( \lambda \)- and \( \nu \)-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 (12a) and (12b), observe that

$$w_{x_i}^x \leq \bar{\kappa}_\ell - \bar{\nu}_\ell \leq \bar{\kappa}_\ell - [\bar{\nu}_\ell]$$

holds for every $\ell \in [q]$. Since $w_{x_i}^x$ is integral, we can round the right-hand side of this inequality down and obtain $w_{x_i}^x \leq [\bar{\kappa}_\ell] - [\bar{\nu}_\ell]$. Thus, (12b) holds for all $\ell \in L$. Moreover, exactly the same arguments can be used to show that $X'$ fulfills (12a) for every $\ell \in L$ and $i \in Z^*_\ell$. To show that $X'$ fulfills (12b) if $\ell \notin L$, we estimate

$$w_{x_i}^x \leq \bar{\kappa}_\ell - \bar{\nu}_\ell = \bar{\kappa}_\ell + [\bar{\nu}_\ell] - [\bar{\nu}_\ell] = \kappa'_\ell - \nu'_\ell.$$ 

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

$$\lambda'_i - \nu'_i = [\bar{\lambda}_i] - [\bar{\nu}_\ell] \geq w_{x_i}^x$$

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

$$Z \ni \lambda'_i - \nu'_i = \bar{\lambda}_i - \vartheta(\bar{\lambda}_i) - (\bar{\nu}_\ell + 1 - \vartheta(\bar{\nu}_\ell)) \geq w_{x_i}^x + \vartheta(\bar{\nu}_\ell) - \vartheta(\bar{\lambda}_i) - 1$$

$$\ell \notin L \quad \Rightarrow \quad w_{x_i}^x - 1 \in \mathbb{Z}.$$ 

This shows that $X'$ fulfills (12a) for all $\ell \notin L$ and $i \in Z^*_\ell$.

Thus, it remains to prove that $X'$ does not violate (12c) for any $\ell \in [q]$. To this end, define $\xi = 1$ if $\ell \notin L$ and $\xi = 0$ otherwise. Then

$$\sum_{r=\ell}^{q} \sum_{i \in Z^*_\ell} (\bar{\lambda}_i - \lambda'_i) + \sum_{r=\ell+1}^{q} (\bar{\kappa}_r - \kappa'_r) - \sum_{r=\ell}^{q} (\bar{\nu}_r - \nu'_r)$$

$$= \sum_{r=\ell}^{q} \left( \sum_{i \in Z^*_\ell} \vartheta(\bar{\lambda}_i) - \vartheta(\bar{\nu}_r) \right) + \sum_{r=\ell+1}^{q} \vartheta(\bar{\kappa}_r) + \sum_{r=\ell}^{q} \sum_{r \notin L} \vartheta(\bar{\lambda}_i)$$

$$+ \sum_{r=\ell+1}^{q} \left( \bar{\kappa}_r - \kappa'_r - \bar{\nu}_r + \nu'_r \right) - \xi (\bar{\nu}_\ell - [\bar{\nu}_\ell])$$

$$\geq \sum_{r=\ell+1}^{q} (\bar{\kappa}_r - \bar{\kappa}_r - [\bar{\nu}_r] + \bar{\nu}_r - \bar{\nu}_r + [\bar{\nu}_\ell]) = 0.$$ 

Consequently,

$$\mu' - \sum_{r=\ell}^{q} \sum_{i \in Z^*_\ell} \lambda'_i - \sum_{r=\ell+1}^{q} \kappa'_r + \sum_{r=\ell}^{q} \nu'_r = \bar{\mu} - \sum_{r=\ell}^{q} \sum_{i \in Z^*_\ell} \lambda_i + \sum_{r=\ell+1}^{q} \lambda'_i - \sum_{r=\ell}^{q} \bar{\kappa}_r$$

$$+ \sum_{r=\ell+1}^{q} (\bar{\kappa}_r - \kappa'_r) + \sum_{r=\ell}^{q} \bar{\nu}_r - \sum_{r=\ell}^{q} (\bar{\nu}_r - \nu'_r)$$

$$\geq \bar{\mu} - \sum_{r=\ell}^{q} \sum_{i \in Z^*_\ell} \lambda_i - \sum_{r=\ell+1}^{q} \bar{\kappa}_r + \sum_{r=\ell}^{q} \bar{\nu}_r \geq w_{x_i}^y,$$

proving that $X'$ fulfills (12c) and thus all constraints of (12). 

Hence, System (11) provides not only a characterization of tuples $(\bar{x}, y)$ that encode vertices of $P^x_\ell$ via an IP formulation, but it already is a complete linear description of the convex hull of these tuples. To obtain a description of $P^x_\ell$ in the original space, we
substitute all occurrences of $\tilde{x}_i$ in (11) in accordance with $\pi$ by
\[ \tilde{x}_i = x_i - 1 + \sum_{r=1}^{\ell} y_r, \]
which turns System (11) into
\[ \sum_{r=1}^{m} y_r \leq 1, \quad (13a) \]
\[ x_i \leq 1, \quad \ell \in [q], i \in Z^*_\ell, \quad (13b) \]
\[ -|Z^*_\ell| \sum_{r=1}^{\ell} y_r - \sum_{i \in Z^*_\ell} x_i \leq -|Z_\ell|, \quad \ell \in [q], \quad (13c) \]
\[ -x_i - \sum_{r=1}^{\ell} y_r \leq -1, \quad \ell \in [q], i \in Z_\ell, \quad (13d) \]
\[ -y_\ell \leq 0, \quad \ell \in [q]. \quad (13e) \]

The new System (13) consists of two kinds of variables the original $x$-variables describing a vector contained in $P \geq \gamma$ and the indicator variables $y$. To obtain a complete linear description of $P \geq \gamma$, we project the feasible region of (13) 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 $y_q, \ldots, y_{k+1}$ from (13) the obtained inequality system is
\[ \sum_{r=1}^{k} y_r \leq 1, \quad (14a) \]
\[ x_{\tilde{z}_\ell} + y_\ell \leq 1, \quad \ell \in [k], \quad (14b) \]
\[ -|Z^*_\ell| \sum_{r=1}^{\ell} y_r - \sum_{i \in Z^*_\ell} x_i \leq -|Z_\ell|, \quad \ell \in [k], \quad (14c) \]
\[ -x_i - \sum_{r=1}^{\ell} y_r \leq -1, \quad \ell \in [k], i \in Z_\ell, \quad (14d) \]
\[ -y_\ell \leq 0, \quad \ell \in [k]. \quad (14e) \]
\[ -|Z^*_\ell| \sum_{r=1}^{k} y_r - \sum_{i \in Z_\ell} x_i + |Z^*_\ell| \sum_{r=k+1}^{\ell} x_{\tilde{z}_r} \leq b^k_\ell - 1, \quad \ell \in [q] \setminus [k], \quad (14f) \]
\[ -x_{\tilde{z}_r} - \sum_{r=1}^{k} y_r \leq \ell - k - 1, \quad \ell \in [q] \setminus [k], i \in Z^*_\ell, \quad (14g) \]
\[ -x_i \leq 0, \quad \ell \in [q] \setminus [k], i \in Z_\ell, \quad (14h) \]
\[ -\sum_{i \in Z_\ell} x_i \leq -1, \quad \ell \in [q] \setminus [k], \quad (14i) \]
\[ x_i \leq 1, \quad \ell \in A^k, \quad (14j) \]

where $A^k := (\bigcup_{\ell=1}^{q} Z^*_\ell) \cup \{ \tilde{z}_j : j \in \{k+1, \ldots, q\} \}$ and $b^k_\ell := |Z^*_\ell| (\ell - k - 1)$.

**Lemma 31.** After eliminating variables $y_q, \ldots, y_{k+1}$, $k \in [q-1]$, in System (13), the resulting system is given by (14).
Proof. We prove this statement inductively. If $k = q$, both System (13) and (14) coincide. Thus, the induction base holds. For the inductive step, we assume that eliminating variables $y_q, \ldots, y_{k+1}$ via FME from System (13) leads to System (14) with parameter $k$. Hence, it remains to show that we obtain System (14) for parameter $k - 1$ if we eliminate $y_k$ from System (14) for parameter $k$.

To this end, we have to combine Inequalities (14a) and (14b), which have a positive coefficient for $y_k$, with Inequalities (14c)–(14g), in which $y_k$ has a negative coefficient. In the following, we compute these combinations, and we explain how the resulting inequality contributes to (14) for parameter $k - 1$.

$$(14a)+(14c): \quad - \sum_{i \in Z_k} x_i \leq -1. \text{ This inequality extends Family (14i).}$$

$$(14a)+(14d): \quad -x_i \leq 0 \text{ for all } i \in Z_k. \text{ These inequalities extend Family (14h).}$$

$$(14a)+(14e): \quad \sum_{r=1}^{k-1} y_r \leq 1. \text{ This inequality is Inequality (14a) in System (14) for parameter } k - 1.$$

$$(14a)+(14f): \quad - \sum_{i \in Z_k} x_i + |Z_k^*| \sum_{r=k+1}^{\ell} x_{z_r} \leq b_\ell^{k-1} - 1 \text{ for all } \ell \in [q] \setminus [k]. \text{ Observe that these inequalities are dominated by the sum of (14i) and multiples of the upper bound constraints } x_{z_r} \leq 1, r \in \{k, \ldots, q\}. \text{ Since both types of constraints are contained in System (14) for parameter } k - 1, \text{ the derived inequalities are redundant.}$$

$$(14a)+(14g): \quad -x_i + \sum_{r=k+1}^{\ell} x_{z_r} \leq \ell - k \text{ for all } i \in Z_k^*. \text{ By the same argument as in the previous case, these inequalities are redundant since } \sum_{r=k+1}^{\ell} x_{z_r} \leq \ell - k \text{ by (14j)} \text{ and } -x_i \leq 0.$$

$$(14b)+(14e): \quad -|Z_k^*| \sum_{r=1}^{k-1} y_r - \sum_{i \in Z_k} x_i + |Z_k^*| x_{\hat{z}_k} \leq -1. \text{ This inequality extends Family (14f).}$$

$$(14b)+(14d): \quad -x_i + x_{\hat{z}_k} - \sum_{r=1}^{k-1} y_r \leq 0 \text{ for all } i \in Z_k. \text{ These inequalities extend Family (14g).}$$

$$(14b)+(14j): \quad x_{\hat{z}_k} \leq 1. \text{ This inequality extends Family (14i).}$$

$$(14b)+(14f): \quad -|Z_k^*| \sum_{r=1}^{k-1} y_r - \sum_{i \in Z_k} x_i + |Z_k^*| \sum_{r=k}^{\ell} x_{z_r} \leq b_\ell^{k-1} - 1 \text{ for every } \ell \in \{k + 1, \ldots, q\}. \text{ These inequalities extend Family (14f).}$$

$$(14b)+(14g): \quad -x_i + \sum_{r=k}^{\ell} x_{z_r} - \sum_{r=1}^{k-1} y_r \leq \ell - k \text{ for all } \ell \in \{k + 1, \ldots, q\}, i \in Z_k^*. \text{ These inequalities extend Family (14g).}$$

Together with the inequalities in System (14) for parameter $k$ that are independent from $k$, the newly generated inequalities define System (14) for parameter $k - 1$, proving the assertion. \hfill \Box

Theorem 32. Let $\gamma \in S_n$ be a monotone and ordered permutation. Then $P_{\gamma}^\geq \gamma$ is completely described by

\[- \sum_{i \in Z_1} x_i + |Z_1^*| \sum_{r=1}^{\ell} x_{z_r} \leq |Z_1^*| (\ell - 1) - 1, \quad \ell \in [q],\]

\[-x_i + \sum_{r=1}^{\ell} x_{z_r} \leq \ell - 1, \quad \ell \in [q], i \in Z_1^*,\]

\[\sum_{i \in Z_1} x_i \geq 1, \quad \ell \in [q],\]

\[0 \leq x_i \leq 1, \quad i \in [n].\]

Proof. By Lemma 31, a complete linear description of $P_{\gamma}^\geq \gamma$ is given by System (14) for parameter $k = 0$. Since Inequalities (14a)–(14e) vanish and the remaining inequalities simplify to the proposed constraints, the assertion follows. \hfill \Box
Theorem 32 shows that $P_\gamma$ 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 \textit{NP}-hard, see Loos [23].

6. Numerical Experience

To evaluate the effect of the techniques developed in this paper, we have implemented three plug-ins for the framework \textsc{scip} [10] to handle symmetries in binary programs via symresacks. Since \textsc{scip} version 5.0 these plug-ins are included in \textsc{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 [10]; the theoretical background for methods of the symresack and orbisack plug-ins is provided in [13].

The \textit{symresack plug-in} 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 $O(n\alpha(n))$, 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. Additionally, the plug-in for a symresack $P_\gamma$ contains a propagation routine for the FD-inequality corresponding to $\gamma$, which runs in linear time.

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 [23]. These specialized constraints are handled by a second plug-in, the so-called \textit{orbisack plug-in}. Similar to the symresack plug-in, it implements a separation routine for minimal cover inequalities and a propagation routine of the corresponding FD-inequality. In contrast to symresacks, the separation routine for orbisacks has a linear running time. 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 MIP. The orbisack plug-in upgrades itself to an \textit{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.

In our experiments, we use \textsc{scip} 5.0.1 as branch-and-bound framework and \textsc{cplex} version 12.7.1 [14] as LP solver. To compute symmetries of the considered instances, we used an internal function of \textsc{scip} that computes generators of a subgroup $\Gamma$ of the symmetry group of a mixed-integer program (MIP) using \textsc{bliss} 0.73 [17]. 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 and symmetries are computed after presolving. Moreover, all reported average numbers are given in shifted geometric mean

$$\prod_{i=1}^{n} \left(t_i + s\right)^{1/n} - s$$
Table 1. Comparison of different symmetry handling variants for Margot’s instances.

<table>
<thead>
<tr>
<th>Setting</th>
<th>#nodes</th>
<th>time</th>
<th>#opt</th>
<th>#act</th>
<th>method-time</th>
<th>sym-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>271</td>
<td>537.84</td>
<td>6</td>
<td>0</td>
<td>0.40</td>
<td>0.00</td>
</tr>
<tr>
<td>OF</td>
<td>58</td>
<td>434.1</td>
<td>1</td>
<td>15</td>
<td>2.62</td>
<td>0.01</td>
</tr>
<tr>
<td>symre</td>
<td>829.41</td>
<td>92.15</td>
<td>13</td>
<td>15</td>
<td>0.80</td>
<td>0.01</td>
</tr>
<tr>
<td>ppsymre</td>
<td>7611.5</td>
<td>82.81</td>
<td>13</td>
<td>15</td>
<td>0.82</td>
<td>0.02</td>
</tr>
</tbody>
</table>

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. 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 SCIP are deactivated. The symre setting adds for each generator $\gamma$ of $\Gamma$ a symresack or orbisack (if $\gamma$ is a composition of 2-cycles) constraint to the MIP. To handle symmetries of $\gamma$, 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 ppsymre setting extends the symre setting by enabling this upgrade. Finally, the OF setting uses the state-of-the-art method orbital fixing, see Margot [24, 25] and Ostrowski [29], to handle symmetries, which is also available in SCIP 5.0.1.

To be able to test the impact of packing and partitioning symresacks on different kinds of instances, we conducted experiments on three test sets: The MIPLIB2010 [21] benchmark test set, 10 instances of a double tennis scheduling problem by Ghoniem and Sherali [9] as well as 15 highly symmetric instances by Margot [24], which are available through the web page [31].

Although all tennis instances contain symmetries that can be handled by packing symresacks, our experiments have shown that using the ppsymre setting leads only to a minor improvement of the performance of the symre setting. Moreover, we detected in only 3 (out of 87) instances of the MIPLIB2010 benchmark test set symmetries that can be handled by packing symresacks. Thus, results for this test set cannot indicate whether symmetry handling inequalities derived via packing symresacks or ordinary symresacks lead to a better performance. For this reason, we do not report on results for the tennis and MIPLIB instances in our analysis below. Instead, we focus on the highly symmetric test set of Margot (summarized in Table 1), which contains 5 instances out of 15 whose symmetries can be handled by packing or partitioning symresacks.

We observe that the time for computing symmetries and executing the symmetry handling methods, i.e., performing orbital fixing or separating and propagating symmetry handling inequalities, is negligible in our experiments. For this reason, we do not report
on these numbers in detail. However, we would like to point out that the method-time is positive in the default setting, because SCIP always initializes the orbital fixing plug-in.

Using the default setting, Margot’s instances are hard to solve for SCIP (it solves only 6 instances within the time limit). Orbital fixing (OF) improves this number to 10 yielding a speed-up of 69.3%. The polyhedral settings, however, clearly dominate orbital fixing because they solve 13 instances and improve the running time by 91.5% (symre) and 92.3% (ppsymre), respectively. In particular, we observe that we can benefit from incorporating packing and partitioning constraints into symmetry handling inequalities, because using ppsymre instead of symre reduces the running time by 10.1%. The number of nodes needed to solve an instance reduces on average by 8.2%. Thus, we are able to further improve the already well performing symre setting by exploiting additional problem structure in symmetry handling, which leads to tighter cutting planes.

In summary, Table 1 shows that checking whether packing/partitioning symresacks are applicable is not costly on general instances – an observation that also holds on the tennis and MIPLIB instances. In particular, incorporating packing and partitioning constraints into symmetry handling inequalities is able to improve the performance of the symre setting on Margot’s instances by 10.1%, respectively. However, the results on the tennis instances show that symmetry handling inequalities exploiting the additional structure of packing and partitioning constraints might not improve the approach via minimal cover inequalities for general symresacks. Consequently, using constrained symresacks instead of ordinary symresacks does not harm the solution process and has, in general, a positive impact on the performance of a branch-and-bound solver.

6.2. Results for Further Symmetric Problems

The goal of this section is to show that exploiting the additional structure of packing and partitioning symresacks improves the performance of branch-and-bound procedures that do not use symmetry handling routines incorporating packing and partitioning constraints in certain applications. To this end, we used 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. [30], 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:

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

\[
\sum_{j=1}^{n} x_{ij} = 1, \quad i \in [m], \quad (15a)
\]

\[
x_{ij} \leq y_j, \quad (i, j) \in [m] \times [n], \quad (15b)
\]

\[
\sum_{i=1}^{m} d_i x_{ij} \leq T y_j + a_j, \quad j \in [n]. \quad (15c)
\]

In this model, \( x_{ij} = 1 \) if and only if operation \( i \) is assigned to room \( j \). Inequality (15a) guarantees that each operation is assigned to exactly one room, and thus, Inequality (15b)
ensures that \( y_j = 1 \) if room \( j \) is used in the schedule. Finally, Inequality (15c) models that \( a_{ij} \) 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 (15) can be permuted arbitrarily without changing the problem structure. To illustrate these symmetries of ORSP, we arrange the variables as entries of a matrix

\[
X := \begin{pmatrix}
  x_{11} & \ldots & x_{1n} \\
  \vdots & \ddots & \vdots \\
  x_{m1} & \ldots & x_{mn} \\
  y_1 & \ldots & y_m \\
  a_1 & \ldots & a_m
\end{pmatrix},
\]

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

\[
\gamma_j = (x_{1j}, x_{1,j+1})(x_{2j}, x_{2,j+1}) \ldots (x_{mj}, x_{m,j+1}),
\]

see [13, Corollary 31]. Consequently, because \( x_{i,j} + x_{i,j+1} \leq 1 \) by Inequality (15a) and non-negativity of \( x \), the symmetries \( \gamma_j \) 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 \( \gamma \) 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 exploits only the symmetries on the \( x \)-variables. To this end, we exploit that packing symresacks admit a small integer programming formulation via the ordering constraints (5), which can be added directly to the problem formulation to handle all symmetries. In the following, we call the basic formulation extended by (5) \texttt{reform-ppsym}. Moreover, we investigate further approaches to handle symmetries directly in the problem formulation, which we discuss in the following.

Since the group \( \Gamma \) containing all room-symmetries permutes the columns of the variable matrix \( X \) arbitrarily, the variable orbits w.r.t. \( \Gamma \) are the rows of \( X \). A classical approach to (partly) handle these symmetries is to pick a variable orbit/row \( i \) of \( X \) and to add the inequalities \( X_{i,j} \geq X_{i,j+1}, j \in [n-1] \) to the problem formulation, see Liberti [22]. Thus, there is a degree of freedom in selecting the variable orbit. In our experiments, we tested two reformulations using this approach. Either we add \( x_{1,i} \geq x_{1,i+1}, i \in [n-1] \) to the original formulation of ORSP (\texttt{reform-orbit-x}) or \( y_i \geq y_{i+1} \) (\texttt{reform-orbit-y}).

Observe that both \texttt{orbit-x} and \texttt{orbit-y} do not take the partitioning constraints on the \( x \)-variables into account. To incorporate this information into a reformulation of ORSP, we use a simple observation by Kaibel and Pfetsch [19] on the vertex structure of partitioning orbitopes: Every vertex \( x \in \{0,1\}^{m \times n} \) of a partitioning orbitope fulfills \( x_{ij} = 0 \) for
PACKING, PARTITIONING, AND COVERING SYMRESACKS 29

Table 2. Comparison of different symmetry handling variants for ORSP.

<table>
<thead>
<tr>
<th>Setting</th>
<th>#nodes</th>
<th>time</th>
<th>#opt</th>
<th>#act</th>
<th>method-time</th>
<th>sym-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>154.641.3</td>
<td>481.12</td>
<td>27</td>
<td>0</td>
<td>0.87</td>
<td>0.00</td>
</tr>
<tr>
<td>OF</td>
<td>84912.6</td>
<td>292.41</td>
<td>30</td>
<td>12</td>
<td>0.95</td>
<td>0.00</td>
</tr>
<tr>
<td>symre</td>
<td>413.9</td>
<td>3.81</td>
<td>49</td>
<td>49</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>reform-orbit-x</td>
<td>82177.5</td>
<td>304.78</td>
<td>35</td>
<td>0</td>
<td>0.61</td>
<td>0.00</td>
</tr>
<tr>
<td>reform-orbit-y</td>
<td>143894.1</td>
<td>453.51</td>
<td>27</td>
<td>0</td>
<td>0.88</td>
<td>0.00</td>
</tr>
<tr>
<td>reform-triangle</td>
<td>8401.9</td>
<td>31.91</td>
<td>49</td>
<td>0</td>
<td>0.13</td>
<td>0.00</td>
</tr>
<tr>
<td>reform-ppsym</td>
<td>614.7</td>
<td>5.20</td>
<td>49</td>
<td>0</td>
<td>0.01</td>
<td>0.00</td>
</tr>
</tbody>
</table>

all $i < j$, i.e., the upper triangle of a vertex is fixed to 0. By adding these variable fixings to the problem formulation (reform-triangle), we can partly remove room-symmetries.

For our experiments, we generated 49 random instances of ORSP with $m = 20$ and $n = 10$ to which we applied SCIP using the settings default, OF, and symre described in the last section. Moreover, we compared the performance of these general symmetry handling settings against the reformulations described above, where all symmetry handling methods are deactivated. A summary of our results is provided by Table 2. Note that Table 2 does not report on results for the setting ppsymre, because SCIP does not (yet) autonomously detect that it suffices to handle only symmetries of $x$-variables. Hence, it does not apply partitioning symresacks due to the above argumentation.

Table 2 shows that although the considered instances are relatively small (220 variables and 230 constraints), they are relatively hard to solve for SCIP without symmetry handling methods or SCIP using orbital fixing. In both settings, SCIP solves only 27 or 30 instances, respectively. Using symresacks, however, SCIP is able to solve all instances within the time limit and on average in less than four seconds. Thus, the results show that symresacks are a powerful tool to handle symmetries in ORSP.

An explanation for the bad performance of orbital fixing on these instances is that SCIP detects many fixings of variables and additional cutting planes within the root node of the branch-and-bound tree. In many of the instances, SCIP incorporates the found cuts and variable fixings into a new model and restarts the solution process on the modified model. Since orbital fixing is not supported after restarts in SCIP 5.0.1, it is reasonable that OF performs poorly because no symmetry handling methods are applied after a restart.

By comparing the different reformulations of ORSP, we see that reform-orbit-y only slightly improves the average running time but keeps the number of solved instances invariant. Thus, ordering only the rooms that are open does not suffice to handle symmetries adequately. Instead, it is more important to handle symmetries on the $x$-variables, i.e., on the operations that are assigned to the rooms, since the remaining reformulations improve the number of solved instances. The simple symmetry handling inequalities used in reform-orbit-x solve 8 additional instances and allow to speed-up the solution process by 36.7%. However, the only reformulations that are competitive with the symre setting are the ones that take the partitioning structure on the $x$-variables into account. In particular, the speed-up of 98.9% achieved by reform-ppsym is comparable to the speed-up of 99.2% of symre, while reform-triangle reduces the solution time only by 93.4%. Thus, room-symmetries can efficiently be handled by modeling partitioning orbitopes directly in the problem formulation – either incompletely by fixing the upper triangle to 0 or completely by adding an IP formulation of packing symresacks to the problem formulation.

In summary, the experiments show that both the approach via unconstrained symresacks using the complete symmetry information and the approach via packing symresacks using only symmetries on the $x$-variables lead to comparable results. The advantage of the latter approach, however, is that it is not relying on a separation routine for the exponentially
many minimal cover inequalities for symresacks. Instead, it suffices to add linearly many
inequalities to the problem formulation to handle all the symmetry of the problem. In
particular, this approach is advantageous if a black box MIP solver is not able to handle
the symmetries efficiently (e.g., SCIP using OF).

Another advantage of constrained symresacks in applications is that the derived sym-
metry handling inequalities can be used even if a MIP solver is not able to detect symmetries.
For example, if a user extends a MIP solver by her own plug-in, the solver might not
be able to detect symmetries. But if the user knows that symmetries handable by pack-
ing/partitioning symresacks are present, she can add the linearly many inequalities of the
IP formulation (5) to the problem formulation to handle symmetries.

Consequently, the ordering constraints of packing and partitioning symresacks provide
an efficient mechanism to handle symmetries directly in the problem formulation if the
user knows that packing or partitioning symresacks are applicable.

6.2.2. Results for Star Coloring Problems. In this section, we report on the effect of sym-
metry handling for the maximum 2-star colorable subgraph problem (M2SCP), which we
have introduced in Example 8. Recall that M2SCP contains two kinds of symmetries:
graph and color symmetries. While color symmetries can easily be handled by packing
orbitopes, the derivation of symmetry handling inequalities for graph symmetries is typ-
ically more complicated and tedious, see, e.g., Januschowski and Pfetsch [15, 16]. 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 8. 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 [12]
by a routine to handle graph symmetries. This method uses Nauty [28] to detect graph
automorphisms and it checks whether the automorphism acts on a clique. To test the im-
 pact of the different variants to handle graph symmetries in M2SCP, we use eight different
settings: In the default setting, no symmetries are handled, whereas the setting colsym
uses a packing orbitope to handle color symmetries. The remaining settings extend the
colsym setting by handling graph symmetries via propagating unconstrained symresacks and

- Us0: not separating unconstrained symresacks,
- Us1: separating unconstrained symresacks,
- Pp1: checking whether packing symresacks are applicable and propagating packing
  symresacks,
- Ps1: checking whether packing symresacks are applicable and separating packing
  symresacks,

which leads to the six settings Us0-Pp0-Ps0, Us1-Pp0-Ps0, Us0-Pp1-Ps0, Us0-Pp1-Ps1,
Us1-Pp1-Ps0, and Us1-Pp1-Ps1.

To test the effect of the different settings, we ran experiments on instances from the
Color02 symposium [4] that contain graph symmetries. Since memory consumption might
be very large if the graphs are big, we only considered the 74 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-pp of instances in Color02 which contain
automorphisms of cliques, i.e., every instance in Color02-pp contains a graph symmetry
that can be handled by a packing symresack. The results of our experiments are provided
in Table 3.

The experiments show that exclusively handling color symmetries leads only to a minor
improvement of the running time by 2.2%. In fact, this is not surprising because there
is only one non-trivial color permutation in M2SCP. Instead, the additional handling of
Table 3. Comparison of different symmetry handling variants for M2SCP.

<table>
<thead>
<tr>
<th>Setting</th>
<th>#nodes</th>
<th>time</th>
<th>#opt</th>
<th>#act</th>
<th>method-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color02 (74):</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>7260.7</td>
<td>294.82</td>
<td>37</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>partsym</td>
<td>6687.4</td>
<td>288.48</td>
<td>37</td>
<td>74</td>
<td>1.57</td>
</tr>
<tr>
<td>Us0-Pp0-Ps0</td>
<td>6135.8</td>
<td>275.61</td>
<td>38</td>
<td>74</td>
<td>1.48</td>
</tr>
<tr>
<td>Us1-Pp0-Ps0</td>
<td>6396.0</td>
<td>277.72</td>
<td>38</td>
<td>74</td>
<td>3.12</td>
</tr>
<tr>
<td>Us0-Pp1-Ps0</td>
<td>5136.9</td>
<td>246.87</td>
<td>41</td>
<td>74</td>
<td>1.07</td>
</tr>
<tr>
<td>Us0-Pp1-Ps1</td>
<td>5195.7</td>
<td>254.05</td>
<td>41</td>
<td>74</td>
<td>1.23</td>
</tr>
<tr>
<td>Us1-Pp1-Pp0</td>
<td>5133.6</td>
<td>244.81</td>
<td>41</td>
<td>74</td>
<td>2.39</td>
</tr>
<tr>
<td>Us1-Pp1-Ps1</td>
<td>5193.3</td>
<td>253.92</td>
<td>41</td>
<td>74</td>
<td>2.58</td>
</tr>
<tr>
<td>Color02-pp (55):</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>5872.2</td>
<td>240.51</td>
<td>29</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>partsym</td>
<td>5819.6</td>
<td>244.30</td>
<td>29</td>
<td>55</td>
<td>1.95</td>
</tr>
<tr>
<td>Us0-Pp0-Ps0</td>
<td>5607.1</td>
<td>238.43</td>
<td>30</td>
<td>55</td>
<td>1.82</td>
</tr>
<tr>
<td>Us1-Pp0-Ps0</td>
<td>5867.9</td>
<td>240.92</td>
<td>30</td>
<td>55</td>
<td>3.50</td>
</tr>
<tr>
<td>Us0-Pp1-Ps0</td>
<td>4409.4</td>
<td>205.92</td>
<td>33</td>
<td>55</td>
<td>1.27</td>
</tr>
<tr>
<td>Us0-Pp1-Ps1</td>
<td>4481.8</td>
<td>213.22</td>
<td>33</td>
<td>55</td>
<td>1.54</td>
</tr>
<tr>
<td>Us1-Pp1-Pp0</td>
<td>4360.5</td>
<td>202.73</td>
<td>33</td>
<td>55</td>
<td>2.52</td>
</tr>
<tr>
<td>Us1-Pp1-Ps1</td>
<td>4415.3</td>
<td>213.25</td>
<td>33</td>
<td>55</td>
<td>2.74</td>
</tr>
</tbody>
</table>

Graph symmetries is more important, since it improves the running time by 5.8%–17.0% and allows to solve between 1 and 4 additional instances. In particular, the experiments show that there is a significant difference between the settings that do not consider the packing structure of graph automorphisms of cliques (Us0-Pp0-Ps0 and Us1-Pp0-Ps0) and the settings that exploit the packing structure. While unconstrained symresacks improve the running time by up to 6.5%, the speed-up caused by the application of packing symresacks varies between 13.8% and 17.0%. Similar observations hold for the subset Color02-pp of instances from Color02 that contain automorphisms of cliques. Consequently, the usage of packing symresacks allows almost to treble the speed-up of unconstrained symresacks.

Observe that separating and propagating unconstrained symresacks and only propagating packing symresacks is most efficient for M2SCP. Thus, although packing symresacks can be characterized by linearly many inequalities of type (5), our experiments indicate that it is more efficient to handle these inequalities indirectly by a propagation routine for M2SCP. In general, however, the separation of Inequalities (5) may have a positive effect on the performance of a MIP solver. For example, (unreported) experiments on the Margot test set show that the running time is smallest when we separate (5) in addition to the propagation of packing and partitioning symresacks.

Summarizing our results for the maximum 2-star colorable subgraph problem, it does not suffice to use general symmetry handling inequalities to handle graph symmetries of M2SCP. Instead, it is necessary to use a problem specific approach to handle symmetries more efficiently. However, the development of problem specific cuts that combine symmetry with additional problem properties is typically tedious and the derived cuts can only be used in a specific application. To avoid the exhaustive analysis of a problem, our experiments have provided evidence that it suffices to consider the general framework of packing and partitioning symresacks to achieve a significant better performance than an approach that does not incorporate problem information into symmetry handling inequalities. In particular, due to the generality of the concept of packing and partitioning symresacks, the application of ordering constraints (5) is not limited to M2SCP, but can be used in many applications.
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. By using the theoretical separation routine, however, one has no control on the size of coefficients of the separated inequalities. Thus, numerical instabilities may occur if such inequalities are used in practice. For this reason, we investigated the special case of packing, partitioning, and covering symresacks in more detail, and we proved that these polytopes admit formulations with bounded coefficients in certain cases. In practice, we have seen that symmetry handling inequalities incorporating packing constraints might have only a moderate impact on the performance of a MIP solver on general instances. In certain applications, however, the derived symmetry handling inequalities are essential for an efficient solving process.

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 an anonymous referee for helpful comments on an earlier version of this paper. Moreover, the author thanks Marc E. Pfetsch for many fruitful discussions.

References


[31] [http://www.mathematik.tu-darmstadt.de/~pfetsch/symmetries.html](http://www.mathematik.tu-darmstadt.de/~pfetsch/symmetries.html).


Research Group Discrete Optimization, Department of Mathematics, Technische Universität Darmstadt, Germany

E-mail address: hojny@opt.tu-darmstadt.de