Math. Program., Ser. A (2019) 175:197–240
https://doi.org/10.1007/s10107-018-1239-7

FULL LENGTH PAPER

Polytopes associated with symmetry handling

Christopher Hojny[1] · **Marc E. Pfetsch**[1]

Received: 26 January 2017 / Accepted: 16 January 2018 / Published online: 24 January 2018

Abstract This paper investigates a polyhedral approach to handle symmetries in mixed-binary programs. We study *symretopes*, i.e., the convex hulls of all binary vectors that are lexicographically maximal in their orbit with respect to the symmetry group. These polytopes turn out to be quite complex. For practical use, we therefore develop an integer programming formulation with ternary coefficients, based on knapsack polytopes arising from a single lexicographic order enforcing inequality. We show that for these polytopes, the optimization as well as the separation problem of minimal cover inequalities can be solved in almost linear time. We demonstrate the usefulness of this approach by computational experiments, showing that it is competitive with state-of-the-art methods and is considerably faster for specific problem classes.

Mathematics Subject Classification 90C09 · 90C11 · 90C57

1 Introduction

In the last decades, a direction of research has investigated methods for solving symmetric mixed integer programs (MIPs). In this case, it is well known that branch-and-bound solvers are slowed down by the fact that symmetric solutions reappear

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10107-018-1239-7) contains supplementary material, which is available to authorized users.

✉ Christopher Hojny
hojny@mathematik.tu-darmstadt.de

Marc E. Pfetsch
pfetsch@mathematik.tu-darmstadt.de

[1] Department of Mathematics, Research Group Optimization, TU Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany

multiple times during the solving process. The aim of symmetry handling methods it to reduce this effect.

More precisely, we consider permutation symmetries of binary programs of the following form:

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

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. Let the *symmetric group* $\mathcal{S}_n$ on $[n] := \{1, \dots, n\}$ operate on $x \in \mathbb{R}^n$ such that a permutation $\gamma \in \mathcal{S}_n$ permutes the coordinates of x to obtain $\gamma(x) := (x_{\gamma^{-1}(1)}, \dots, x_{\gamma^{-1}(n)})^\top$. A subgroup Γ of $\mathcal{S}_n$, denoted by $\Gamma \le \mathcal{S}_n$, is a *symmetry group* if for each $\gamma \in \Gamma$ and $x \in \{0,1\}^n$

- $A\gamma(x) \le b$ if and only if $Ax \le b$ (invariance of feasibility) and
- $c^\top \gamma(x) = c^\top x$ if $Ax \le b$ (invariance of the objective).

For different definitions of symmetry groups and an overview of symmetry handling we refer the reader to Margot [43].

A well established way to handle symmetry is by adding inequalities that cut off symmetric solutions. In fact, this idea has been used in many specific applications, e.g., Sherali and Smith [51], Ostrowski et al. [44], Ghoniem and Sherali [17], Faenza and Kaibel [14], and [32]. More general methods were investigated by Friedman [16], Liberti [34,35], Liberti and Ostrowski [36], and Dias and Liberti [12]. Most of these approaches define a lexicographic order on the set of solutions and apply inequalities to cut off solutions that are not lexicographically maximal in their orbit.

This suggests the very natural polyhedral approach of this paper: we investigate *symretopes*, the convex hulls of lexicographically maximal representative systems of solutions. In fact, all of the aforementioned symmetry handling methods that depend on a lexicographic ordering generate valid inequalities for symretopes. Moreover, this approach provides the strongest polyhedral formulation of a lexicographically maximal representative set if only the symmetry group is taken into account and no further problem specific information.

The investigation of symretopes, which are formally defined in Sect. 2, proceeds as follows. We study their polyhedral properties in Sect. 2.1, and we give some examples for symretopes for which a complete linear description is available in Sect. 2.2. Our results show that the group structure of Γ has a strong impact on the polyhedral properties and interacts in complex ways with the lexicographic ordering. In fact, symretopes can be very complicated and complete descriptions are not available, in general. Thus, in order to exploit the symmetry handling effect, IP-formulations become important. In fact, Friedman [16] gave a formulation based on fundamental domains. Unfortunately, it contains exponentially large coefficients, making it impractical.

One aim is therefore to derive an IP-formulation with small coefficients. To this end, in Sect. 3, we consider knapsack polytopes, so-called *symresacks*, defined by specific symmetry breaking inequalities. It turns out that the structure of symresacks is less complex compared to symretopes. In fact, we prove that the linear optimization problem over symresacks can be solved efficiently in Sect. 3.1, while it is $\mathcal{NP}$-hard for symretopes. Moreover, in Sect. 3.2 we describe a tractable IP-formulation for symretopes by exploiting the efficiently solvable separation problem of minimal cover inequalities for symresacks. In particular, this IP-formulation can be separated in time polynomial in the dimension of the problem and the size of the considered group. In Sect. 3.3, we show how to improve this IP-formulation for special symresacks, so-called orbisacks. Furthermore, the polyhedral structure of symresacks allows for characterizations of facet defining properties of basic inequalities, see Sect. 3.4.

The computational impact of the derived formulations is investigated in Sect. 4. We test two variants of the above IP-formulations and show that they are competitive with state-of-the-art methods and much more effective on certain test sets.

Literature overview. Further polyhedral symmetry handling techniques are discussed in the literature. For instance, a polyhedral approach to reduce symmetries in integer programs is based on core points, introduced by Herr et al. [23,24]. The *orbit* of a solution x is the set $\Gamma(x) := \{\gamma(x) \;:\; \gamma \in \Gamma\}$ that contains all permutations of x w.r.t. Γ. A *core point* is a point $x \in \mathbb{Z}^n$ such that the only integral points in the convex hull of its orbit $\Gamma(x)$ are the points in the orbit themselves. If x is a feasible point, it suffices to consider only core points, because integer points in the convex hull of $\Gamma(x)$ have the same objective value as x. This idea was successfully applied to problems of small dimensions whose symmetry groups are products of symmetric groups, see Herr et al. [23]. For a detailed discussion of core points, we refer the reader to Herr [22] and Rehn [48].

Unfortunately, the restriction to core points for *binary* programs does not reduce the number of treated solutions in branch-and-bound solvers, since each feasible point x is a core point. Symretopes, however, can be used to handle symmetries in this case.

Besides polyhedral approaches, a variety of other symmetry handling methods for MIPs are discussed in the literature. Most of these techniques aim to consider at most one representative of each orbit of a solution during the solving process: *Isomorphism pruning*, see Margot [39–42], prunes a node v of the branch-and-bound tree if it detects that no solution in the subtree rooted in v contains a solution that is lexicographically maximal in its orbit. Another symmetry handling technique for binary programs proposed by Ostrowski et al. [46] suggests to reduce symmetry by a branching rule that either fixes one element in an orbit to 1 or all elements to 0, so-called *orbital branching*. A modification of this approach for specially structured symmetry groups was discussed by Ostrowski et al. [45]. Furthermore, *orbital shrinking* by Fischetti and Liberti [15] aims to exploit symmetry by introducing artificial variables that represent the variable orbits. Instead of the original problem, they suggest to solve a reduced ("shrunk") problem in the new orbit variables.

2 Symmetry breaking polytopes

In this section, we introduce symmetry breaking polytopes for permutation groups $\Gamma \leq \mathcal{S}_A$ on a finite set A, i.e., subgroups of the symmetric group $\mathcal{S}_A$ on A. If $A = [n] := \{1, \ldots, n\}$, we write $\mathcal{S}_n$ instead of $\mathcal{S}_{[n]}$. For the remainder of this paper, we assume that n is an integer with $n \geq 2$.

Definition 1 The *symmetry breaking polytope (symretope)* for $\Gamma \leq \mathcal{S}_n$ is

$$\mathrm{S}(\Gamma) := \mathrm{conv}(\{x \in \{0,1\}^n : \bar{c}^\top x \geq \bar{c}^\top \gamma(x) \quad \forall\, \gamma \in \Gamma\}),$$

where $\bar{c} = (2^{n-1}, 2^{n-2}, \ldots, 2, 1)^\top \in \mathbb{R}^n$ is the *universal ordering vector*.

Friedman [16] proved that the binary points which are contained in the cone generated by the inequalities

$$\bar{c}^\top x \geq \bar{c}^\top \gamma(x), \quad \gamma \in \Gamma, \tag{1}$$

are exactly the lexicographic maximal points of the orbits $\Gamma(x)$, $x \in \{0,1\}^n$. Thus, the vertices of $\mathrm{S}(\Gamma)$ form a lexicographic maximal representative system of binary vectors w.r.t. the group Γ. We call the inequalities (1) *fundamental domain (FD) inequalities*, since they define a fundamental domain for the action of Γ on $\mathbb{R}^n$; see Coxeter [10] for an introduction to the concept of fundamental domains. Consequently, we can handle symmetry in binary programs completely by adding the FD-inequalities for each permutation in the symmetry group of the binary program, since only those solutions remain feasible that are lexicographically maximal in their orbit. If x is lexicographically not smaller than y, we denote this by $x \succeq y$; if x is lexicographically greater than y, we write $x \succ y$.

Using FD-inequalities to handle symmetry in binary programs may not be favorable from a computational point of view: On the one hand, if Γ contains exponentially many permutations, we have to add exponentially many inequalities to handle symmetries. On the other hand, the entries of $\bar{c}$ grow exponentially in n, and thus, may lead to numerical instabilities. Of course, one can use different ordering vectors than $\bar{c}$ in (1) and Definition 1. But then, the number of vertices of $\mathrm{S}(\Gamma)$ (and thus, the number of feasible points in the binary program) may increase, see Friedman [16, Thm. 2]. For this reason, we restrict our attention to the universal ordering vector in this paper.

Unfortunately, we can not expect to find a tractable description of symretopes for general groups as the optimization problem over symretopes is $\mathcal{NP}$-hard if Γ is given by a set of generators:

Proposition 2 *Let $w \in \mathbb{R}^n$ and let $\Gamma \leq \mathcal{S}_n$. It is $\mathcal{NP}$-hard to maximize $w^\top x$ over $x \in \mathrm{S}(\Gamma)$, if Γ is given by a set of generators.*

Proof Babai and Luks [3] proved that for $x \in \{0,1\}^n$ it is $\mathcal{NP}$-hard to find a permutation $\gamma \in \Gamma$ such that $\gamma(x)$ is lexicographically maximal along its Γ-orbit, if Γ is given by a set of generators. Consequently, the membership problem for $\mathrm{S}(\Gamma)$, and thus, the optimization problem, is $\mathcal{NP}$-hard, see Grötschel et al. [18]. □

Not surprisingly, the polyhedral structure of symretopes turns out to be very complicated. Indeed, this is even the case for the cyclic group $\mathcal{C}_n$. Although the action of $\mathcal{C}_n$ is very simple and there are several algorithms to generate the vertices of $\mathrm{S}(\mathcal{C}_n)$ via the theory of Lyndon words, see, e.g., Chen et al. [7] or Sawada and Williams [49], finding a complete linear description of $\mathrm{S}(\mathcal{C}_n)$ remains open. For this reason, we study basic polyhedral properties of symretopes in the following. This will allow us to derive in Sect. 3 an IP-formulation for symretopes of polynomial size, which removes symmetries from a binary program by inequalities with coefficients in $\{0, \pm 1\}$, if the size of the considered permutation group is polynomially bounded in the dimension. In particular, these results are applicable to cyclic groups.

2.1 Polyhedral properties of symretopes

In this section, we investigate properties of the polyhedral structure of symretopes. In the following, we denote the characteristic vector of a set $A \subseteq [n]$ by χ^A, where $\chi_i^A = 1$ if $i \in A$ and $\chi_i^A = 0$ otherwise.

Lemma 3 *Let* $\Gamma \leq \mathcal{S}_n$*. Then,* $\dim(\mathrm{S}(\Gamma)) = n$*, i.e.,* $\mathrm{S}(\Gamma)$ *is full-dimensional.*

Proof Consider the characteristic vectors $\chi^{[k]}$, $k \in [n]_0 := \{0\} \cup [n]$. These affinely independent vectors are lexicographically maximal in their orbit, since a permutation does not change the number of 1-entries. Hence, $\mathrm{S}(\Gamma)$ is full-dimensional. □

Lemma 4 *If* $\Gamma' \leq \Gamma \leq \mathcal{S}_n$*, then* $\mathrm{S}(\Gamma) \subseteq \mathrm{S}(\Gamma')$.

Proof If Γ' is a subgroup of Γ, $\mathrm{S}(\Gamma')$ is defined via a subset of constraints of $\mathrm{S}(\Gamma)$. □

In the following, we take the group structure into account, and we start with a result on product groups. Let $A_1 = [n_1]$ and $A_2 = [n] \setminus A_1$ be a partition of $[n]$. Moreover, let $\Gamma_i \leq \mathcal{S}_{A_i}$, $i \in [2]$. The action of any $\gamma_i \in \Gamma_i$, $i \in [2]$, is naturally extended to $[n]$ by leaving elements outside of A_i invariant. Then elements $(\gamma_1, \gamma_2) \in \Gamma_1 \otimes \Gamma_2$ of the direct product act on $[n] = A_1 \cup A_2$ by applying each γ_i on A_i.

Figure 1a shows the symretope of $\mathcal{S}_{[2]}$ if considered as a subgroup of $\mathcal{S}_2$, whereas Fig. 1b illustrates the symretope for $\mathcal{S}_{[2]}$ as a subgroup of $\mathcal{S}_3$. In the latter case, $\mathrm{S}(\mathcal{S}_{[2]}) \subseteq [0, 1]^3$ is a prism with ground face $\mathrm{S}(\mathcal{S}_{[2]}) \subseteq [0, 1]^2$. This is not by chance, since $\Gamma := \mathcal{S}_{[2]} \leq \mathcal{S}_3$ can be written as $\Gamma \otimes \{\mathrm{id}_{\mathcal{S}_{\{3\}}}\}$ where id_Γ is the identity permutation of Γ. In the following proposition, we generalize this observation:

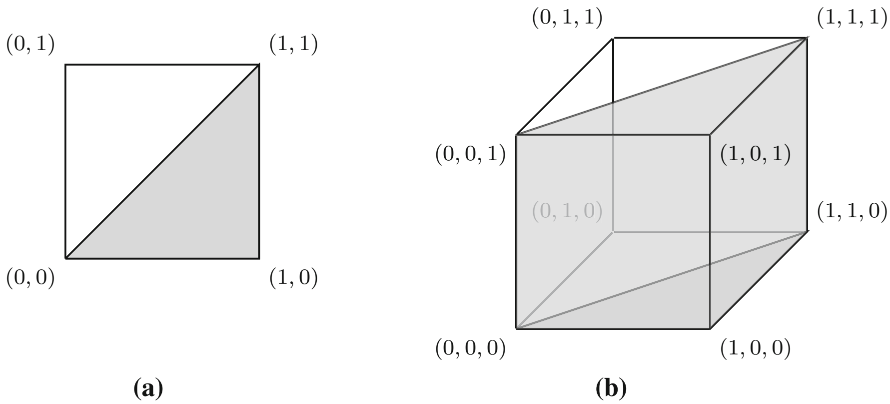

Fig. 1 Symretopes for $\mathcal{S}_{[2]}$ considered as a subgroup of $\mathcal{S}_2$ and $\mathcal{S}_3$. The symretopes are given as the gray part of the two- and three-dimensional unit cube, respectively. **a** $\mathrm{S}(\mathcal{S}_{[2]}) \subseteq [0, 1]^2$, **b** $\mathrm{S}(\mathcal{S}_{[2]}) \subseteq [0, 1]^3$

Proposition 5 *Let $A_1 \cup A_2$ be a partition of $[n]$ with $A_1 = [n_1]$, $A_2 = [n] \setminus A_1$, and let $\Gamma_i \leq \mathcal{S}_{A_i}$, $i \in [2]$. If $\Gamma = \Gamma_1 \otimes \Gamma_2$, then* $\mathrm{S}(\Gamma) = \mathrm{S}(\Gamma_1) \times \mathrm{S}(\Gamma_2)$.

Proof Since $\Gamma = \Gamma_1 \otimes \Gamma_2$, we know that Γ contains all permutations of the form $\gamma = (\gamma_1, \mathrm{id}_{\Gamma_2})$ and $\gamma' = (\mathrm{id}_{\Gamma_1}, \gamma_2)$ where $\gamma_i \in \Gamma_i$, $i \in [2]$. Thus,

$$\begin{aligned} \mathrm{S}(\Gamma) &\subseteq (\mathrm{S}(\Gamma_1) \times [0,1]^{A_2}) \cap ([0,1]^{A_1} \times \mathrm{S}(\Gamma_2)) \\ &= \mathrm{S}(\Gamma_1) \times \mathrm{S}(\Gamma_2) =: S^\star. \end{aligned}$$

Hence, it remains to prove that also the reverse inclusion holds.

Observe that the order which is induced by $\bar{c} \in \mathbb{R}^n$ restricted onto A_i is the lexicographic order on A_i, $i \in [2]$. Thus, if $x \in S^\star$, x is lexicographically maximal in its orbit restricted to A_i. Otherwise, there would be a permutation $\gamma_i \in \Gamma_i$ that fixes all entries of x not in A_i and that permutes x onto an element which is lexicographically greater on A_i, and hence, $\bar{c}^\top x < \bar{c}^\top \gamma_i(x)$. Since each $\gamma \in \Gamma$ can be written as the product of permutations $(\gamma_1, \mathrm{id}_{\Gamma_2})$, $\gamma_1 \in \Gamma_1$, and $(\mathrm{id}_{\Gamma_1}, \gamma_2)$, $\gamma_2 \in \Gamma_2$, this implies that we cannot increase the ordering value $\bar{c}^\top x$ by permuting x by any $\gamma \in \Gamma$. Consequently, $S^\star \subseteq \mathrm{S}(\Gamma)$. □

Remark 6 Proposition 5 can be generalized easily to more than two factors and to non-consecutive sets of a partition of $[n]$. Thus, if Γ is an arbitrary product group, we can handle all symmetries of Γ in a binary program by adding valid inequalities/facet defining inequalities for the symretopes of the factors of Γ.

Because $\mathrm{S}(\Gamma)$ is a 0/1-polytope, it is natural to analyze whether the valid inequalities $x_i \geq 0$ and $x_i \leq 1$, $i \in [n]$, define facets of $\mathrm{S}(\Gamma)$. To this end, we exploit homogeneity of Γ. The action of $\Gamma \leq \mathcal{S}_n$ is *k-homogeneous* if for each $A_1, A_2 \subseteq [n]$ with $|A_1| = |A_2| = k$, there exists a permutation $\gamma \in \Gamma$ such that $A_2 = \gamma(A_1)$. If $k = 1$, we call the action of Γ *transitive*.

Lemma 7 *Let $\Gamma \leq \mathcal{S}_n$. Then, $x_n \geq 0$ and $x_1 \leq 1$ define facets of* $\mathrm{S}(\Gamma)$. *If Γ acts transitively on $[n]$, $x_2 \leq 1, \ldots, x_n \leq 1$ do* not *define facets of* $\mathrm{S}(\Gamma)$.

Proof Because $\chi^{[k]}$, $k \in [n-1]_0$, are affinely independent vertices of $\mathrm{S}(\Gamma)$, $x_n \geq 0$ defines a facet of $\mathrm{S}(\Gamma)$. Furthermore, the characteristic vectors $\chi^{[k]} \in \mathrm{S}(\Gamma)$, $k \in [n]$, fulfill $x_1 = 1$ and are affinely independent. Thus, $x_1 \leq 1$ defines a facet of $\mathrm{S}(\Gamma)$.

Observe that each binary vector in $\mathrm{S}(\Gamma)$ with $x_i = 1$ also fulfills $x_1 = 1$ if Γ acts transitively on $[n]$ for $i \geq 2$: Assume for the sake of contradiction, there exists $x \in \mathrm{S}(\Gamma) \cap \{0,1\}^n$ with $x_1 = 0$ and $x_i = 1$. By transitivity, there exists $\gamma \in \Gamma$ such that $\gamma(i) = 1$ and thus, $\gamma(x)_1 = 1$. Hence, $\gamma(x) \succ x$ which is a contradiction. Thus, each vertex of $\mathrm{S}(\Gamma)$ which is contained in the face induced by $x_i = 1$ is also contained in the face induced by $x_1 = 1$ if Γ acts transitively. Hence, $x_i \leq 1$ cannot define a facet if $i > 1$, since $\mathrm{S}(\Gamma)$ is full-dimensional. □

If $\Gamma \leq \mathcal{S}_n$ does not act transitively, there may exist a fixed point $i \in [n]$. In this case, $\Gamma \cong \Gamma' \otimes \mathrm{id}_{\{i\}}$ for some $\Gamma' \leq \mathcal{S}_{n-1}$, and thus, $\mathrm{S}(\Gamma)$ is linearly equivalent to the Cartesian product $\mathrm{S}(\Gamma') \times [0,1]$, cf. Proposition 5. Consequently, $x_i \leq 1$ and $x_i \geq 0$ define facets for every fixed point. For the remaining points, we obtain the following:

Lemma 8 *Let $\Gamma \leq \mathcal{S}_n$, $\Gamma \neq \{\mathrm{id}_\Gamma\}$, and let $j \in [n]$ be the smallest integer that is not a fixed point of Γ. Then $x_j \leq 1$ defines a facet of* $\mathrm{S}(\Gamma)$ *and $x_i \leq 1$, $i \in \Gamma(j)\setminus\{j\}$, does not define a facet.*

Proof W.l.o.g. we can assume that $j = 1$, because fixed points do not affect lexicographical maximality of a solution w.r.t. index permutations. Then $x_j \leq 1$ defines a facet by Lemma 7. Moreover, $x_i \leq 1$, $i \in \Gamma(j)\setminus\{j\}$ cannot define a facet of $\mathrm{S}(\Gamma)$, because Γ acts transitively on $\Gamma(j)$. Thus, the same arguments as in Lemma 7 apply to show the second part of the assertion. □

Remark 9 Let j be defined as in Lemma 8. Analyzing the arguments in the proofs of Lemmas 7 and 8 shows that $-x_j + x_i \leq 0$, $i \in \Gamma(j)\setminus\{j\}$, is a valid inequality for $\mathrm{S}(\Gamma)$ that together with $x_j \leq 1$ dominates $x_i \leq 1$. These inequalities were already described in the literature, see Liberti [35, Prop. 15]. Note, however, that these inequalities need not define a facet of $\mathrm{S}(\Gamma)$ for every $i \in \Gamma(j)$, see, e.g., Proposition 15 below for $\Gamma = \mathcal{S}_n$. However, one can show that $x_1 \geq x_i$ defines a facet for all $i \in \{2, \ldots, \lfloor \frac{n}{2} \rfloor + 1\}$ of $\mathrm{S}(\mathcal{C}_n)$.

A characterization of groups for which the trivial inequalities define facets seems to be complex: For example, if Γ is the group that is generated by the permutation $(1, 6, 5, 4, 2, 3)$, the inequalities $x_i \geq 0$, $i \in \{3, 4, 5, 6\}$, define facets of $\mathrm{S}(\Gamma)$. But if Γ is the group that is generated by $(1, 2, 3, 4, 5, 6)$, the only non-negativity inequality which defines a facet of $\mathrm{S}(\Gamma)$ is $x_6 \geq 0$. Since both groups act transitively, this shows that transitivity does not rule out that $x_i \geq 0$ for $i \in [n-1]$ define facets of $\mathrm{S}(\Gamma)$.

Lemma 7 and its proof show that for transitive groups $x_1 \leq 1$ contains each non-zero vertex of $\mathrm{S}(\Gamma)$. We thus have the following geometric result.

Corollary 10 *Let $\Gamma \leq \mathcal{S}_n$ be a group acting transitively on $[n]$. Then, $\mathrm{S}(\Gamma)$ is a pyramid with the origin as apex. Consequently, if $a^\top x \leq b$ is a facet defining inequality for $\mathrm{S}(\Gamma)$ which is not $x_1 \leq 1$, then $b = 0$.*

Since the cyclic group $\mathcal{C}_n$ acts transitively on $[n]$, $\mathrm{S}(\mathcal{C}_n)$ is a pyramid by Corollary 10.

In the introduction of Sect. 2 we mentioned that it may not be favorable to use FD-inequalities to handle symmetry due to their exponentially large coefficients. In fact, only FD-inequalities with small coefficients can define facets of symretopes as we will see in the next proposition. Observe, however, that there are symretopes whose facets need coefficients of exponential size, see Sect. 2.2.2. Thus, the reason why only FD-inequalities with small coefficients can define facets of $\mathrm{S}(\Gamma)$ does not lie in the size of coefficients but the vertex structure of symretopes.

In the following, we write $\Gamma^\star$ instead of $\Gamma\setminus\{\mathrm{id}_\Gamma\}$.

Proposition 11 *For each $\gamma \in \Gamma^\star$ which is not a transposition, the corresponding FD-inequality (1) does not define a facet of $\mathrm{S}(\Gamma)$. If γ is the next neighbor transposition $(i, i+1)$ for some $i \in [n-1]$, then the FD-inequality defines a facet of $\mathrm{S}(\Gamma)$.*

Proof Let $i \in [n-1]$ such that $(i, i+1) \in \Gamma$. Then, the corresponding FD-inequality is given by $-x_i + x_{i+1} \leq 0$, since

$$\begin{aligned}\bar{c}^\top x \geq \bar{c}^\top \gamma(x) &\Leftrightarrow 2^{n-i}\, x_i + 2^{n-i-1}\, x_{i+1} \geq 2^{n-i}\, x_{i+1} + 2^{n-i-1}\, x_i\\ &\Leftrightarrow 2\, x_i + x_{i+1} \geq 2\, x_{i+1} + x_i.\end{aligned}$$

The characteristic vectors $\chi^{[k]} \in \mathrm{S}(\varGamma)$, $k \in [n]_0$, $k \neq i$, fulfill this inequality with equality, yielding n affinely independent vectors. Thus, the above inequality defines a facet.

On the other hand, consider $\gamma \in \varGamma^\star$ which is not a transposition. By definition, the FD-inequality for γ is given by

$$\bar{c}^\top \gamma(x) - \bar{c}^\top x \leq 0 \quad \Leftrightarrow \quad \underbrace{\sum_{i=1}^{n} 2^{n-i}\, x_{\gamma^{-1}(i)}}_{\text{sum 1}} - \underbrace{\sum_{i=1}^{n} 2^{n-i}\, x_i}_{\text{sum 2}} \leq 0. \tag{2}$$

Let $\zeta_1 \circ \cdots \circ \zeta_m$ be the disjoint cycle decomposition of γ, where we include the trivial cycles (j) for each fixed point $j \in [n]$ of γ. Since γ is not a transposition and $\gamma \neq \mathrm{id}_\varGamma$, $m < n - 1$ holds. Assume that (2) defines a facet of $\mathrm{S}(\varGamma)$. Then, there exists $\hat{x} \in \mathrm{S}(\varGamma) \cap \{0, 1\}^n$ which fulfills (2) with equality such that $\hat{x}$ is not constant on $\{i \in [n] \,:\, \zeta_j(i) \neq i\}$ for some $j \in [m]$: Otherwise, the affine hull of the face F which is induced by (2) would be affinely equivalent to $\mathbb{R}^m$. Consequently, there would be at most $m + 1 < n$ affinely independent vectors in F, contradicting the assumption that (2) defines a facet.

By assumption, $\hat{x} \in F \cap \{0, 1\}^n$ is not a fixed point of γ. Hence, the index $k := \min\{i \in [n] \,:\, \hat{x}_i \neq \hat{x}_{\gamma^{-1}(i)}\}$ is well-defined. Observe that $\hat{x}_k = 1$ as well as $\hat{x}_{\gamma^{-1}(k)} = 0$, since otherwise $\gamma(\hat{x}) \succ \hat{x}$. Then, -2^{n-k} is the smallest coefficient with corresponding $\hat{x}_k = 1$ which is not canceled out by its positive counterpart in sum 1 of (2). Because $\hat{x}$ fulfills (2) with equality, this implies

$$0 = \sum_{i=k+1}^{n} 2^{n-i}\, \hat{x}_{\gamma^{-1}(i)} - \sum_{i=k+1}^{n} 2^{n-i}\, \hat{x}_i - 2^{n-k} \leq (2^{n-k} - 1) - 0 - 2^{n-k} = -1.$$

Thus, we obtain a contradiction and the inequality cannot define a facet. □

Remark 12 If $\gamma = \mathrm{id}_\varGamma$, $\varGamma \leq \mathcal{S}_n$, the corresponding FD-inequality is $0 \geq 0$, and thus redundant.

Observe that the second part of Proposition 11 cannot be strengthened to "if and only if", since there are groups for which an arbitrary transposition (i, j) induces a facet, e.g., the group $\{\mathrm{id}_{[4]}, (1, 3), (2, 4), (1, 3)(2, 4)\}$ for which the FD-inequalities for both transpositions define facets. Moreover, there are groups such that the corresponding symretope is completely described by the trivial inequalities and FD-inequalities, see Proposition 15 below.

Note that the inequality $-x_i + x_{i+1} \leq 0$ is an FD-inequality for the next neighbor transposition $(i, i + 1)$. However, we show next that it defines a facet of $\mathrm{S}(\varGamma)$ even if $(i, i+1) \notin \varGamma$, but $\varGamma$ has a sufficient homogeneity order. In particular, these inequalities define facets of the symretope for the *alternating group* $\mathcal{A}_n$, which is the group of all permutations in $\mathcal{S}_n$ that can be written as a composition of an even number of transpositions, since it acts k-homogeneously on $[n]$ for each $k \in [n-2]$. Thus, the FD-inequalities for next neighbor transpositions defines facets of $\mathrm{S}(\mathcal{A}_n)$ although $\mathcal{A}_n$ does not contain any transposition.

Proposition 13 *Let $\Gamma \leq \mathcal{S}_n$ be a group acting k-homogeneously on $[n]$ for $k \in [\lceil \frac{n}{2} \rceil]$. Then $-x_i + x_{i+1} \leq 0$ defines a facet of $\mathrm{S}(\Gamma)$ for all $i \in [k]$.*

Proof By Proposition 11, the inequality $-x_i + x_{i+1} \leq 0$ defines a facet of $\mathrm{S}(\mathcal{S}_n)$ for each $i \in [n-1]$ and by Lemma 4, $\mathrm{S}(\mathcal{S}_n) \subseteq \mathrm{S}(\Gamma)$. Hence, it suffices to show that $-x_i + x_{i+1} \leq 0$ is valid for $\mathrm{S}(\Gamma)$ if $i \in [k]$, because $\mathrm{S}(\mathcal{S}_n)$ is full-dimensional and thus $\dim(\mathrm{S}(\Gamma)) = \dim(\mathrm{S}(\mathcal{S}_n))$.

Since Γ acts k-homogeneously on $[n]$, Γ acts also ℓ-homogeneously for $\ell \leq k \leq \lceil \frac{n}{2} \rceil$, see Dixon and Mortimer [13]. Hence, for all binary vectors $x \in \mathrm{S}(\Gamma)$ with $\mathbb{1}^\top x = \ell$, $\ell \in [k]$, it follows that $x_j = 1$ if $j \in [\ell]$ and $x_j = 0$ otherwise. Furthermore, if $\mathbb{1}^\top x > k$, the first k entries in x have to be 1-entries by the same argument. Hence, the inequalities $-x_i + x_{i+1} \leq 0$ are valid for $\mathrm{S}(\Gamma)$. □

Unfortunately, almost all finite permutation groups act at most 4-homogeneously on $[n]$, see Dixon and Mortimer [13, p. 34 and Thm. 9.4B]. Thus, Proposition 13 shows that $-x_i + x_{i+1} \leq 0$ can only define a facet if $i \in [4]$ in most cases. An important exception is the symmetric group $\mathcal{S}_n$ that acts k-homogeneously on $[n]$ for every $k \in [n]$.

We conclude this section with a comment on the number of vertices of $\mathrm{S}(\Gamma)$.

Remark 14 By a classical result in Pólya theory, see, e.g., van Lint and Wilson [54], one can determine the number of vertices of arbitrary symretopes: Denote by $c_k(\Gamma)$ the number of permutations in $\Gamma \leq \mathcal{S}_n$ which have exactly k cycles in their disjoint cycle decomposition. Then the number of vertices of $\mathrm{S}(\Gamma)$ is exactly $\frac{1}{|\Gamma|} \sum_{k=1}^{\infty} c_k(\Gamma)\, 2^k$. In particular, $\mathrm{S}(\mathcal{C}_n)$ has exactly $\frac{1}{n} \sum_{d|n} \varphi(d)\, 2^{n/d}$ vertices, where $\varphi \colon \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$ is Euler's phi function.

2.2 Examples

This section deals with examples for symretopes of particular groups and describes cases in which a complete linear description of symretopes is known. Thus, if these groups are symmetry groups of binary programs, we can handle all symmetries of these binary programs by strong inequalities for symretopes. Afterwards, we concentrate on groups for which we can at least derive a compact extended formulation.

2.2.1 The symmetric and alternating group

In this section, we investigate the symretopes for the symmetric and alternating group.

Proposition 15 *The symretopes $\mathrm{S}(\mathcal{S}_n)$ and $\mathrm{S}(\mathcal{A}_n)$, $n \geq 3$, are completely described by the inequalities $-x_i + x_{i+1} \leq 0$ for all $i \in [n-1]$ as well as $x_1 \leq 1$ and $-x_n \leq 0$.*

Proof Denote with S the polytope which is defined by the given inequalities. Because these inequalities are FD-inequalities of next neighbor transpositions, they are valid for $\mathrm{S}(\mathcal{S}_n)$. Moreover, we observe that the inequalities $-x_i + x_{i+1} \leq 0$ and $-x_{i+1} + x_{i+2} \leq 0$ are valid for the alternating group $\mathcal{A}_{\{i,i+1,i+2\}}$, $i \in [n-2]$, which is generated by the permutation $(i, i+1, i+2)$. Since all 3-cycles are contained in $\mathcal{A}_n$, the validity of the given inequalities for $\mathrm{S}(\mathcal{A}_n)$ follows. Hence, $\mathrm{S}(\mathcal{A}_n),\ \mathrm{S}(\mathcal{S}_n) \subseteq S$.

Furthermore, the induced polytope S is integral: The constraint matrix of the inequalities $-x_i + x_{i+1} \leq 0$ is totally unimodular, since it is the incidence matrix of a directed graph. Finally, each vertex of S is of the form $\chi^{[k]}$ for some $k \in [n]$, because the inequalities $-x_i + x_{i+1} \leq 0$ enforce that the entries are ordered non-increasingly, and thus, the remaining inequalities ensure that $x \in [0, 1]^n$. Since these vectors are lexicographically maximal in their orbit w.r.t. $\mathcal{S}_n$, we conclude that $S = \mathrm{S}(\mathcal{S}_n)$. Moreover, this implies $\mathrm{S}(\mathcal{A}_n) = S$ by Lemma 4. □

Remark 16 Proposition 15 shows that $\mathrm{S}(\mathcal{S}_n)$ and $\mathrm{S}(\mathcal{A}_n)$ are simplices with vertices $\chi^{[k]}$ for $k \in [n]_0$. In particular, this shows that symretopes for different groups can coincide.

2.2.2 Orbitope actions

If we do not consider the action of $\Gamma \leq \mathcal{S}_n$ on $\mathbb{R}^n$ but an induced action, the situation is different: Let $\Gamma \leq \mathcal{S}_n$ act on $\{0, 1\}^{m \times n}$, the set of all binary $(m \times n)$-matrices, by permuting the columns. In the literature, the convex hulls of binary matrices whose columns are sorted lexicographically non-increasing w.r.t. Γ are known as *full orbitopes*, see [32]. To make the concept of full orbitopes compatible with symretopes, we assume that the universal ordering vector $\bar{c}$ assigns the value $2^{mn-(i-1)n-j}$ to the entry (i, j) of an $(m \times n)$-matrix.

For full orbitopes w.r.t. $\Gamma = \mathcal{S}_n$, a compact extended formulation is available, see Kaibel and Loos [29], but a complete linear description remains open. Furthermore, a complete linear description of packing and partitioning orbitopes is known, see [32]. These are full orbitopes w.r.t. $\Gamma = \mathcal{S}_n$ that have the additional property that each vertex has at most/exactly one 1-entry per row, respectively. For example, these polytopes naturally appear in graph coloring problems, see [27,32] for a detailed discussion of this example.

In the remainder of this paper, we will frequently refer to properties of full orbitopes w.r.t. the symmetric group $\mathcal{S}_2$, so-called *orbisacks*:

$$\mathrm{O}_m := \mathrm{conv}(\{X = (X^1, X^2) \in \{0, 1\}^{m \times 2} \ : \ X^1 \succeq X^2\}),$$

i.e., the convex hull of all binary $(m \times 2)$-matrices X whose first column X^1 is lexicographically not smaller than its second column X^2. Kaibel and Loos [30] gave an extended formulation as well as a complete linear description, and Loos [37] presented an algorithm for optimization over orbisacks. The complete linear description has $\Theta(3^m)$ facets and there is a facet inequality whose largest coefficient is 2^{m-2}. Moreover, Kaibel and Loos observed the following structure of vertices of O_m which will be useful in Sect. 3.2:

Observation 1 (Kaibel and Loos [30]) Let X be a vertex of an orbisack. Either each row of X is (0, 0) or (1, 1), so-called *constant rows*, or there exists a so-called *critical row* c which has a (1, 0)-pattern and all rows i above c, i.e., $i < c$, are constant, whereas below row c an arbitrary 0/1-pattern is allowed.

2.2.3 Wreath products

Other actions on 0/1-matrices are given by wreath products of two groups $\Lambda \leq \mathcal{S}_m$ and $\Gamma \leq \mathcal{S}_n$. The natural action of the wreath product $\Gamma \wr \Lambda$ on 0/1-matrices $X \in \mathbb{R}^{m \times n}$ is to permute the order of the rows according to Λ and each row vector by a (possibly distinct) permutation in Γ. In particular, we can permute the entries of each row of X independently from the entries in other rows by a permutation in Γ.

Wreath products appear, for example, as symmetry groups of specially composed graphs G, see Harary [20], and thus, as a symmetry group of the binary program which describes the coloring problem of G. An example for the wreath product of symmetric groups as the symmetry group of an assignment problem is the fully social golfer problem, see Harvey [21].

In the following, we derive an extended formulation for $\mathrm{S}(\Gamma \wr \mathcal{S}_m) \subseteq \mathbb{R}^{m \times n}$, where $\Gamma \leq \mathcal{S}_n$. As aforementioned, we assume that the universal ordering vector $\bar{c}$ assigns entry (i, j) of a matrix in $\{0, 1\}^{m \times n}$ the value $2^{mn-(i-1)n-j}$. Furthermore, we denote with $X_{i\cdot}$ the ith row of a matrix. With this convention we have the following property.

Lemma 17 *A matrix $X \in \{0, 1\}^{m \times n}$ is a vertex of $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$ if and only if*

- *$X_{i\cdot}$, $i \in [m]$, is a vertex of $\mathrm{S}(\Gamma)$ and*
- *the rows of X are sorted lexicographically non-increasing, i.e., $X_{i\cdot} \succeq X_{i+1\cdot}$ for each $i \in [m-1]$.*

Proof Let $X \in \{0, 1\}^{m \times n}$. If there is a row of X which is not a vertex of $\mathrm{S}(\Gamma)$, there is a permutation $\gamma \in \Gamma \wr \Lambda$ which permutes this row to a vertex of $\mathrm{S}(\Gamma)$ and which keeps the other rows invariant. Due to the definition of the ordering vector $\bar{c}$, the matrix $\gamma(X)$ is lexicographically greater than X. Similarly, if each row of X is a vertex of $\mathrm{S}(\Gamma)$, but the rows of X are not sorted lexicographically non-increasing, there is a permutation $\lambda \in \mathcal{S}_m$ which sorts the rows of X lexicographically non-increasing. This implies $\lambda(X) \succ X$, and consequently, each binary matrix that is lexicographically maximal in its orbit w.r.t. $\Gamma \wr \mathcal{S}_m$ has the proposed properties. Thus, the mentioned properties are necessary.

To prove sufficiency, let $X \in \{0, 1\}^{m \times n}$ fulfill both conditions. If X was not a vertex of $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$, there would exist a reordering $\lambda \in \mathcal{S}_m$ of the rows of X as well as permutations $\gamma_i \in \Gamma$, $i \in [m]$, that permute the entries of row i of X such that the permutation $\bar{\gamma} := \lambda \circ \gamma_1 \circ \cdots \circ \gamma_m$ (seen as an element of $\mathcal{S}_{[m]\times[n]}$ in the obvious way) fulfills $\bar{\gamma}(X) \succ X$. Observe, however, that γ_i has to leave the ith row of X invariant, because $X_{i\cdot}$ is the (unique) lexicographically maximal element in its orbit and the rows of each vertex of $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$ have to be lexicographically maximal in their Γ-orbit by the necessity of the conditions. Thus, $\bar{\gamma} = \lambda$ holds. But this is a contradiction, because $\bar{\gamma}(X) \succ X$ implies that the rows of X cannot be sorted lexicographically non-increasing. Consequently, the sufficiency of both conditions follows. □

Lemma 17 immediately leads to an extended formulation of $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$. Let V be the set of vertices of $\mathrm{S}(\Gamma)$. We introduce for each row $i \in [m]$ and each vertex $v \in V$ a variable y_i^v which shall indicate that row i of a vertex $X \in \mathrm{S}(\Gamma \wr \mathcal{S}_m)$ coincides with v. Moreover, we consider the following inequalities

$$\sum_{v \in V} y_1^v \leq 1, \tag{3}$$

$$\sum_{v \in V} y_m^v \geq 1, \tag{4}$$

$$\sum_{\substack{w \in V: \\ w \succeq v}} y_{i+1}^w \leq \sum_{\substack{w \in V: \\ w \succeq v}} y_i^w, \quad \forall\, (i, v) \in [m-1] \times V, \tag{5}$$

which together with the non-negativity inequalities $y_i^v \geq 0$, $(i, v) \in [m] \times V$, define the polytope $S \subseteq [0, 1]^{[m] \times V}$.

Proposition 18 *Let $\Gamma \leq \mathcal{S}_n$. The symretope $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$ is the projection of the integral polytope S under $\pi : \mathbb{R}^{[m] \times V} \to \mathbb{R}^{m \times n}$,*

$$y \mapsto \pi(y), \quad \text{with} \quad \pi(y)_{i\cdot} = \sum_{v \in V} y_i^v\, v^\top, \qquad \forall i \in [m].$$

Proof Let V be the vertex set of $\mathrm{S}(\Gamma)$ and define the directed graph $G = (V_G, A_G)$ by $V_G = ([m] \times V) \cup \{(0, s),\ (m+1, t)\}$ and

$$\begin{aligned} A_G := &\big\{\big((i, v),\ (i+1, w)\big) \ :\ i \in [m-1],\ v, w \in V,\ v \succeq w\big\} \\ &\cup \big\{\big((0, s),\ (1, v)\big),\ \big((m, v),\ (m+1, t)\big) \ :\ v \in V\big\}, \end{aligned}$$

see Fig. 2 for an illustration. In the following, we will identify the node $(0, s)$ with s and $(m+1, t)$ with t. Lemma 17 implies that there exists a bijection between the s-t-paths in G and the vertices X of $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$ via the mapping that assigns row i of X vertex v if and only if (i, v) is in the s–t-path.

Define variables y_i^v for all $(i, v) \in V_G$. Vande Vate [55] showed that the convex hull of all incidence vectors of nodes in s–t-paths in G is given by

$$\begin{aligned} y_0^s = y_{m+1}^t &= 1, \\ \sum_{(i,v) \in T} y_i^v - \sum_{(i,v) \in \mathrm{pred}(T)} y_i^v &\leq 0, \qquad \forall\, T \subseteq V_G \setminus \{(0, s)\}, \\ y_i^v &\geq 0, \qquad \forall\, (i, v) \in V_G, \end{aligned} \tag{6}$$

where $\mathrm{pred}(T)$ denotes all predecessors of T in G. Consequently, $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$ is the image of the s–t-path polytope for G under the map π. Observe that both y_0^s and y_{m+1}^t are fixed to 1, and thus, the unfixed variables of the s–t-path polytope coincide with the variables of S. Hence, it suffices to show that the s–t-path polytope (without considering y_0^s and y_{m+1}^t as variables) is S.

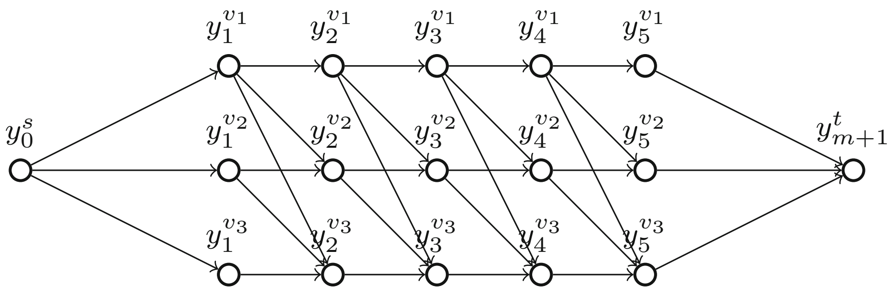

Fig. 2 Illustration of the graph G in the proof of Proposition 18 with $|V| = 3$ and $m = 5$. We assume that $v^i \succ v^{i+1}$ for each $i \in [4]$

Let the vertex set of $\mathrm{S}(\Gamma)$ be $V = \{v^1, \ldots, v^k\} \subseteq \{0, 1\}^n$ with $v^1 \succ v^2 \succ \cdots \succ v^k$. Observe that for $T = \{(1, v) : v \in V\}$, Inequality (6) is $\sum_{v \in V} y_1^v \leq y_0^s = 1$. For $T = \{(m+1, t)\}$, Inequality (6) is given by $1 = y_{m+1}^t \leq \sum_{v \in V} y_m^v$. Moreover, for the set $T_\ell^i := \{(i, v^j) : j \in [\ell]\}$, $(i, \ell) \in [m] \times [|V|]$, the inequality is

$$\sum_{\substack{w \in V: \\ w \succeq v}} y_{i+1}^w \leq \sum_{\substack{w \in V: \\ w \succeq v}} y_i^w.$$

Thus, these inequalities coincide with (3)–(5). To prove the assertion, it remains to show that all remaining inequalities (6) are redundant.

If $T \subseteq \{(i, v) : v \in V\}$ for some $i \in [m]$ and if v^ℓ is the lexicographically smallest element in a tuple $(i, v) \in T$, we can assume that $T = T_\ell^i$: Otherwise, Inequality (6) for T can be dominated by the inequality for T_ℓ^i, since $\mathrm{pred}(T) = \mathrm{pred}((i, v^\ell)) = \mathrm{pred}(T_\ell^i)$. Similarly, if $T = T_\ell^i \cup T_{\ell'}^{i'}$ for some $(i, \ell), (i', \ell') \in [m] \times [|V|]$ with $i \neq i'$, then Inequality (6) for T can be dominated by the sum of the inequalities (6) for the sets T_ℓ^i and $T_{\ell'}^{i'}$.

Using these arguments, we can conclude by induction on the number of sets in the partition $T = \bigcup_{i=1}^m T^i$, $T^i \subseteq \{(i, v^j) : j \in [|V|]\}$, that Inequality (6) is redundant if $T \neq T_\ell^i$ for some $(i, \ell) \in [m] \times [|V|]$. This concludes the proof. □

Remark 19 Alternatively, one can easily prove Proposition 18 by applying the criterion of Ghouhila–Houri, see, e.g., Schrijver [50], to show that the constraint matrix (3)–(5) is totally unimodular.

The extended formulation of Proposition 18 shows that $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$ has a polynomial extension complexity if the number of vertices of $\mathrm{S}(\Gamma)$ is bounded by a polynomial in n; for instance, if $\Gamma = \mathcal{S}_n$ the number of vertices is $n + 1$. For a definition of the extension complexity and an overview on extended formulations, see Kaibel [28] and Conforti et al. [8].

In the case where Γ is the alternating group $\mathcal{A}_n$ or the symmetric group $\mathcal{S}_n$, we can describe $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$ completely by linear inequalities in the original variable space by exploiting Proposition 15.

Proposition 20 *If $\Gamma = \mathcal{A}_n$ or $\Gamma = \mathcal{S}_n$, the symretope* $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$ *is completely described by the following irredundant inequality system:*

$$-X_{i,j} + X_{i,j+1} \leq 0, \qquad \forall\, i \in [m],\ j \in [n-1], \tag{7}$$
$$-X_{i,j} + X_{i+1,j} \leq 0, \qquad \forall\, i \in [m-1],\ j \in [n], \tag{8}$$
$$X_{1,1} \leq 1, \tag{9}$$
$$-X_{m,n} \leq 0. \tag{10}$$

Proof Note that $X_{i,1} \leq 1$ is equal to the sum of inequalities of type (8) and Inequality (9). Similarly, $-X_{i,n} \leq 0$ is the sum of inequalities of type (8) and (10). Hence, these inequalities and (7) guarantee that the ith row of an $(m \times n)$-matrix X is a vertex of $\mathrm{S}(\Gamma)$, see Proposition 15. By Lemma 17, it remains to enforce that the rows of X are sorted lexicographically non-increasing. Because the vertices of $\mathrm{S}(\Gamma)$ are the vectors $\chi^{[k]}$, $k \in [n]_0$, this is implied by (8). Thus, each integer point which fulfills (7)–(10) is a vertex of $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$. But since (7) and (8) form the incidence matrix of a directed graph, the constraint matrix corresponding to (7)–(10) is totally unimodular. Consequently, (7)–(10) is a complete linear description of $\mathrm{S}(\Gamma \wr \mathcal{S}_m)$. □

Summarizing this section, there are complete linear descriptions for symretopes for symmetric and alternating groups as well as for wreath products of symmetric groups. For full orbitopes, we know a polynomial size extended formulation, while a complete linear description in the original space is unknown. In fact, these group actions cover a significant part of symmetries appearing in practice: the paper [47] contains a list of all symmetries of the 194 instances of MIPLIB 2010 [33], which may have a non-trivial symmetry group. Analyzing this list, it turns out that 93 instances contain symmetries of a full symmetric group and 7 instances contain symmetries of wreath products of symmetric groups; the symmetries of 80 instances can be handled by full orbitopes. In total, the symmetries of 108 instances of MIPLIB 2010 can be completely handled by (valid inequalities for) symretopes for these three types.

Nevertheless, symretopes are well understood only for very exclusive symmetry groups and one may wonder about complete linear descriptions for further groups that are mathematically interesting, for example, the cyclic group. Note, however, that no cyclic groups (other than $\mathcal{S}_2$) appear in the MIPLIB 2010.

3 Symresacks

In the last section, we have seen several examples of symretopes for which a complete linear description is available, but in general, complete descriptions of arbitrary symretopes are unknown. In this section, we focus on 0/1-knapsack polytopes that are induced by a *single* FD-inequality, so-called symresacks. We will later see why they are knapsack polytopes and how they can be used to obtain an IP-formulation for symretopes. In particular, the latter will allow us to handle symmetries in arbitrary binary programs by inequalities with ternary coefficients.

Definition 21 The *symresack for* $\gamma \in \mathcal{S}_n$ is the polytope

$$\begin{aligned}\mathrm{P}_\gamma &:= \operatorname{conv}(\{x \in \{0,1\}^n \;:\; \bar{c}^\top x \geq \bar{c}^\top \gamma(x)\}) \\ &= \operatorname{conv}(\{x \in \{0,1\}^n \;:\; x \succeq \gamma(x)\}).\end{aligned}$$

First, we analyze the optimization problem over symresacks. Second, we exploit our analysis of the optimization problem for arbitrary symresacks to develop an efficient algorithm to solve the separation problem of minimal cover inequalities for symresacks. As a consequence, we derive a tractable IP-formulation of $\mathrm{S}(\Gamma)$ with coefficients in $\{0, \pm 1\}$ if $|\Gamma|$ is bounded by a polynomial in n. Thus, the corresponding set of (exponentially many) inequalities can be separated efficiently. Third, we consider orbisacks, which are special symresacks, and we improve the separation algorithm for minimal cover inequalities in this case. Finally, we investigate basic polyhedral properties of symresacks and present some examples.

3.1 An almost linear time optimization algorithm for symresacks

In the following, we analyze the problem of maximizing a linear objective over a symresack. In contrast to symretopes, it will turn out that over symresacks this optimization problem can be solved in $\mathcal{O}(n\,\alpha(n))$ time, where $\alpha(n)$ is the inverse Ackermann function. Our approach builds on Observation 1 on the structure of vertices of orbisacks, which can be used to prove that the linear optimization problem over P_γ is polynomially solvable as follows. Define the polytope

$$\mathrm{Q}_\gamma := \operatorname{conv}(\{X = (X^1, X^2) \in \{0,1\}^{n\times 2} \;:\; X^1 \succeq X^2,\; X^2 = \gamma(X^1)\}),$$

i.e., the convex hull of all vertices of the orbisack O_n such that the second column is a permutation of the first column according to γ.

Lemma 22 *For* $\gamma \in \mathcal{S}_n$*,* P_γ *and* Q_γ *are linearly equivalent.*

Proof Let x be a vertex of P_γ. Then x is lexicographically not smaller than $\gamma(x)$. Because γ is a linear transformation of $\{0,1\}^n$, each vertex of P_γ is transformed via the linear map $\phi\colon \mathrm{P}_\gamma \to \mathrm{Q}_\gamma, x \mapsto (x, \gamma(x))$ onto a vertex of Q_γ. Its inverse $\phi^{-1}\colon\, Q_\gamma \to P_\gamma, \phi^{-1}(x,\gamma(x)) = x$, maps each vertex $(x,\gamma(x))$ of Q_γ into a vertex of P_γ since $x \succeq \gamma(x)$. Consequently, both polytopes are linearly equivalent. □

Theorem 23 *For* $\gamma \in \mathcal{S}_n$*, the linear optimization problem over* P_γ *can be solved in* $\mathcal{O}(n\,\alpha(n))$ *time.*

Note that we use a RAM model for the computations, in particular, elementary arithmetic operations take constant time.

Proof In the following, we describe an algorithm which maximizes a linear objective $W \in \mathbb{R}^{n\times 2}$ over Q_γ. To maximize $w \in \mathbb{R}^n$ over P_γ, we can apply this algorithm to $W = (w, \mathbb{0})$ due to Lemma 22. An example for the application of this algorithm is presented in "Appendix A" of this paper.

Since $\mathrm{Q}_\gamma \subseteq \mathrm{O}_n$ is a 0/1-polytope, all vertices of Q_γ are also vertices of O_n. Hence, either all rows of a vertex X of Q_γ are constant, or there is a critical row $c \in [n]$ of X such that all rows above c are constant and $(X^1_c, X^2_c) = (1, 0)$, see Observation 1. Consequently, if we compute for each $c \in [n+1]$ a vertex that maximizes W over all vertices of Q_γ with critical row c, we obtain an optimal solution by taking one of the computed solutions with maximal objective value. We assume that all rows have to be constant rows if $c = n + 1$.

Let X be a vertex of Q_γ with critical row $c \in [n]$. Thus, $(X^1_c, X^2_c) = (1, 0)$. Then there are further entries in X which have to be 0 or 1 due to the choice of the critical row c: For $i \in [c]$, define $k(i)$ and $\ell(i)$ to be the smallest positive integers such that $\gamma^{k(i)}(i) \geq c$ and $\gamma^{-\ell(i)}(i) \geq c$ (or ∞ if no such integer exists), where $\gamma^{-\ell} := (\gamma^{-1})^\ell$. Furthermore, define

$$F^1_i := \left\{(i, 1), (\gamma(i), 2), (\gamma(i), 1), \ldots, (\gamma^{k(i)-1}(i), 1), (\gamma^{k(i)}(i), 2)\right\},$$
$$F^2_i := \left\{(i, 2), (\gamma^{-1}(i), 1), (\gamma^{-1}(i), 2), \ldots, (\gamma^{-\ell(i)+1}(i), 2), (\gamma^{-\ell(i)}(i), 1)\right\},$$

and $F_i := F^1_i \cup F^2_i$.

Claim If $\gamma^{k(c)}(c) = c$, row c cannot be critical. Otherwise, for critical row $c \leq n$, F^1_c and F^2_c contain all entries of X which have to be 1 or 0 due to the choice of c, respectively. Furthermore, if $i \in [c-1]$ such that $(i, 1) \notin F_c$, F_i contains all entries of X that have to obtain the same value as X^1_i.

Proof (of claim) Since $X^2 = \gamma(X^1)$ and row c is critical, $X^1_c = X^2_{\gamma(c)} = 1$. If $\gamma(c) < c$, $X^1_{\gamma(c)} = 1$ has to hold, because all rows above c have to be constant. We can iterate this argument until $\gamma^k(c) \geq c$ for some $k \geq 1$. If $\gamma^k(c) = c$, we would have $X^2_c = 1$, which is a contradiction because row c is critical. Contrary, if $\gamma^k(c) > c$, this has no implications on the value of $X^1_{\gamma^k(c)}$, because rows below c do not have to be constant. Similarly, this argument applies for implications induced by $X^2_c = 0$, but we have to replace γ by γ^{-1}. Consequently, F^1_c and F^2_c contain all entries of X which are 1 and 0 due to the choice of the critical row c, respectively.

Let $i \in [c-1]$ such that $(i, 1) \notin F_c$, i.e., X^1_i corresponds to an entry which is not already fixed to 0 or 1 due to the choice of c. Then, we use a similar argumentation as above to see that F_i contains all entries of a vertex $X \in \mathrm{Q}_\gamma$ with critical row c which have to obtain the same value as X^1_i. $\square$

Observe that F^1_i and F^2_i coincide if $F_i \subseteq [c-1] \times [2]$, i.e., the orbit of $(i, 1)$ lies above c. This does not produce a contradiction, because all entries in F_i obtain the same value.

Claim Let $F, F' \in \{F_1, \ldots, F_c\}$. Then $F = F'$ or $F \cap F' = \varnothing$.

Proof (of claim) Consider the directed graph $G = (V, A)$ with $V = [n] \times [2]$ and arc set $A = \left\{\big((i, 2), (i, 1)\big), \big((i, 1), (\gamma(i), 2)\big) : i \in [n]\right\}$. The cycles in the disjoint cycle decomposition of γ correspond to the strongly connected components of G. Moreover, each node has in- and out-degree one, and the nodes $(i, 1)$ and $(i, 2)$, are contained in the same cycle of G for each $i \in [n]$. By construction, the subgraph W_i which is induced by F_i is connected, and thus a path or a cycle. If W_i is a path, its end nodes are of the form $(u, 1)$, $(v, 2)$ with $u, v \geq c$. Note that these nodes cannot be inner nodes of any $W_{i'}$, $i' \in [c]$. Thus, if two subgraphs W_i, $W_{i'}$ intersect, one being a path, they have to coincide. If W_i, $W_{i'}$ are intersecting cycles, they have to coincide by construction. The latter argument is the same as in the classical result that the orbits of a group action are disjoint. ◻

We can now find a vertex of Q_γ with critical row c that maximizes W by the following procedure. For each $c \in [n+1]$, we construct an undirected graph $G_c = (V_c, E_c)$ with node set $V_c = [n]$ and edge set $E_c = \big\{\{i, \gamma^{-1}(i)\} : i < c\big\}$. The nodes of the connected component of G_c that contains node $i < c$ is equal to set F_i^1 or F_i^2 restricted to entries of the first column of X by definition of F_i^1 and F_i^2. Moreover, the nodes in the connected components of c and $\gamma^{-1}(c)$ correspond to the restriction of the sets F_c^1 and F_c^2 to entries of the first column of X. Finally, if $i > c$ is not contained in a connected component of a node $i' \leq c$, the condition $\gamma(i) > c$ holds. Thus, row i of a vertex of Q_γ with critical row c is neither constant nor critical. Observe that in this case the connected component of node i consists of a single node.

Consequently, a maximizer $X = (X^1, X^2)$ over Q_γ with critical row $c < n+1$ is given by assigning $X_i^1 = X_{\gamma(i)}^2 = 1$ for all i in the connected component of c and $X_i^1 = X_{\gamma(i)}^2 = 0$ for all i in the connected component of $\gamma^{-1}(c)$. If $c = n+1$, we can skip this step because there is no critical row. Afterwards, we iterate over the remaining connected components and assign $X_i^1 = X_{\gamma(i)}^2 = 1$ if i is contained in a connected component C whose weight $\sum_{i \in C}(W_i^1 + W_{\gamma(i)}^2)$ is positive and $X_i^1 = X_{\gamma(i)}^2 = 0$ if the weight of the connected component is non-positive. Note that we can treat the sets F_i, and thus the connected components of G_c, separately because of Claim 3.1. Thus, this procedure is valid and it runs in $\mathcal{O}(n)$ time for each critical row $c \in [n+1]$, yielding a quadratic optimization algorithm for Q_γ since there are $\mathcal{O}(n)$ possible critical rows.

To improve the running time of the algorithm, we generate G_c, $c \in [n+1]$, from the graph G_{c-1}, where $G_0 = (V_0, E_0)$ with $V_0 = [n]$ and $E_0 = \varnothing$. We keep track of the weights of its connected components, because these weights determine the decision whether the entries of the first column of a maximizer are fixed to 0 or 1, see above. Since G_c and G_{c-1} differ by exactly one edge, either two connected components C_1 and C_2 of G_{c-1} are merged into a connected component $C = C_1 \cup C_2$ or all connected components remain the same. In the first case, the weight of C is given by adding the weights of C_1 and C_2, whereas in the second case the weights of all connected components remain unchanged.

Using this observation leads to Algorithm 1, which returns the critical row index of a maximizer over Q_γ: During the c-th iteration of the algorithm, $\mathcal{C}$ contains the connected components of G_c, see Lines 1 and 9, and $\bar{W}$ stores the sum of the weights of all connected components whose weight is positive, see Lines 2 and 10. Since all entries of X^1 that are contained in the same connected component have to obtain the same value in any vertex of Q_γ, the value $\bar{W}$ is the maximal value of a vertex of Q_γ with critical row c, if we ignore the fixings of variables induced by the critical row. To incorporate these implied fixings, in Line 6, we compute $\hat{W}$ which corrects the value of $\bar{W}$, if the variables in the connected component of $\gamma^{-1}(c)$ are not fixed to 0 or the

Algorithm 1: Optimization algorithm for Q_γ

input : $\gamma \in \mathcal{S}_n$, $W \in \mathbb{R}^{n \times 2}$
output: critical row index of a maximizer of W over Q_γ
1 initialize $\mathcal{C} \leftarrow \{\{1\}, \{2\}, \dots, \{n\}\}$ and weights $W(C) \coloneqq \sum_{i \in C} W_i^1 + W_{\gamma(i)}^2$, $C \in \mathcal{C}$;
2 set $\bar{W} \leftarrow \sum_{C \in \mathcal{C}} \max\{W(C), 0\}$, $W^\star \leftarrow -\infty$, and $c^\star \leftarrow 0$;
3 **for** $c \leftarrow 1, \dots, n+1$ **do**
4   let C_1 and C_2 be the sets of $\mathcal{C}$ containing c and $\gamma^{-1}(c)$, respectively;
5   **if** $C_1 \neq C_2$ **then**
6         $\hat{W} \leftarrow \bar{W} - \max\{W(C_2), 0\} + \min\{W(C_1), 0\}$;
7   **if** $\hat{W} > W^\star$ **then**
8             $W^\star \leftarrow \hat{W}$ and $c^\star \leftarrow c$;
9   remove C_1 and C_2 from $\mathcal{C}$, add $C = C_1 \cup C_2$ to $\mathcal{C}$, and set the weight of C to $W(C_1) + W(C_2)$;
10         $\bar{W} \leftarrow \bar{W} - \max\{W(C_1), 0\} - \max\{W(C_2), 0\} + \max\{W(C), 0\}$;
11 return $c^\star$;

variables in the connected component of c are not fixed to 1. Finally, Lines 7 and 8 ensure that we store the critical index c that allows for the maximal objective value.

Consequently, we can find the critical row index of a maximizer of W by calling Algorithm 1. This algorithm can be implemented to run in $\mathcal{O}(n\,\alpha(n))$ time by using a disjoint-set data structure that allows to perform Lines 4 and 9 in $\mathcal{O}(\alpha(n))$ time, see Tarjan [52], Tarjan and van Leeuwen [53], or Cormen et al. [9]. Afterwards, we can construct a maximizer with critical row $c^\star$ in linear time as described above. Thus, the whole optimization procedure has running time of $\mathcal{O}(n\,\alpha(n))$. ◻

Proposition 24 *The extension complexity of* P_γ *for* $\gamma \in \mathcal{S}_n$ *is in* $\mathcal{O}(n^2)$*, i.e., there is an extended formulation of* P_γ *which has* $\mathcal{O}(n^2)$ *inequalities and variables.*

Proof Let P_γ^c, $c \in [n+1]$, be the convex hull of all vertices x of P_γ such that $X \coloneqq (x, \gamma(x))$ has critical row c. Observe that $\mathrm{P}_\gamma^c = \varnothing$ if no vertex with critical row c exists. Then, $\mathrm{P}_\gamma = \mathrm{conv}(\bigcup_{c=1}^{n+1} \mathrm{P}_\gamma^c)$. Denote with s_c the minimal number of inequalities and equations in a complete linear description of P_γ^c, and define $s \coloneqq \max\{s_c : c \in [n+1]\}$. Due to Balas [5] there is an extended formulation of P_γ of size $\mathcal{O}(s\,n)$. Hence, it remains to show that P_γ^c can be described by at most $\mathcal{O}(n)$ inequalities and equations to prove the assertion. Note that we can ignore any $\mathrm{P}_\gamma^c = \varnothing$, $c \in [n+1]$, in the above description. For this reason, it suffices to consider P_γ^c for a fixed critical row $c \in [n]$ in the following. Note that all sets that are defined below depend on the value of c, but we do not indicate this explicitly to keep the notation simple.

We partition the row set $[n]$ of X with critical row c into the following sets. The sets $F^r = \{k \in [n] : (k, 1) \in F_c^r\}$, $r \in [2]$, (with F_c^r defined as in the proof of Theorem 23) contain the entries of the first column of X which are fixed to 1 and 0 due to the choice of the critical row c. Let $A^j = \{k \in [n] : (k, 1) \in F_j\}$, $j \in [c-1]$, i.e., A^j contains the indices of variables of the first column of X which have to be set to the same value due to fixings above the critical row c; recall that $A^j = A^\ell$ is possible for $j \neq \ell$. Let R be a set of representatives of the sets A^j. Then if $r \in R \cap A^j$

and X_r^1 is fixed to $\alpha \in \{0, 1\}$, all other entries in A^j have to be fixed to α, too. Finally, let $B \coloneqq [n] \setminus (F^1 \cup F^2 \cup \bigcup_{j<c} A^j)$, i.e., B contains the rows of variables in the first column of X below the critical row that can be chosen independently. Then, P_γ^c is completely described by the following system, because the constraint matrix is totally unimodular:

$$\begin{aligned} x_k &= 1, && \forall\, k \in F^1, \\ x_k &= 0, && \forall\, k \in F^2, \\ x_k &= x_r, && \forall\, k \in [n],\ r \in R \cap A^k, \\ x_k &\in [0, 1], && \forall\, k \in [n], \end{aligned}$$

which uses $\mathcal{O}(n)$ inequalities and equations. □

In Sect. 2 we have seen that the optimization problem over symretopes $\mathrm{S}(\Gamma)$ is $\mathcal{NP}$-hard in general. But because of Theorem 23, we are now able to prove that the optimization problem over $\mathrm{S}(\Gamma)$ can be solved in polynomial time if the size of the group is constant, i.e., the optimization problem over $\mathrm{S}(\Gamma)$ is fixed parameter tractable in $|\Gamma|$.

Theorem 25 *For each $\Gamma \leq \mathcal{S}_n$, the linear optimization problem over* $\mathrm{S}(\Gamma)$ *can be solved in $\mathcal{O}(|\Gamma|\, n^{|\Gamma|})$ time.*

Proof The symretope $\mathrm{S}(\Gamma)$ can be reformulated as

$$\mathrm{S}(\Gamma) = \mathrm{conv}\Big(\bigcap_{\gamma \in \Gamma^\star} \{x \in \{0, 1\}^n \,:\, \bar{c}^\top x \geq \bar{c}^\top \gamma(x)\} \Big). \tag{11}$$

In the following, we use the algorithm with quadratic running time from Theorem 23, call it Algorithm $\mathcal{A}$, for each $\gamma \in \Gamma^\star$ to obtain an algorithm which solves the optimization problem over $\mathrm{S}(\Gamma)$. Let $\kappa_\gamma \in [n+1]$ be the index of the "critical row" for the symresack associated with $\gamma \in \Gamma^\star$. Furthermore, define for all $i \in [n]$ an indicator $L^i \in \{0, \pm 1\}$, which is initialized with -1. The aim of the indicator is to keep track of the value of x_i in any solution with critical row distribution $\kappa \in \mathbb{R}^{\Gamma^\star}$: If x_i has no fixed value in any solution, $L^i = -1$. Otherwise, L^i corresponds to the value of x_i that was determined by one of the symresacks P_γ.

For each $\gamma \in \Gamma^\star$, we set $x_i = 1$ or $x_i = 0$ if $\mathcal{A}$ detects that each vertex of P_γ with critical row κ_γ takes this value, unless infeasibility is detected. If $L^i = 1 - x_i$, we terminate since another permutation detected that entry i must obtain the value $1 - x_i$, a contradiction; hence, κ cannot encode the critical row structure of a vertex of $\mathrm{S}(\Gamma)$. Otherwise, if $L^i = -1$, we set $L^i = x_i$; then upon termination, L^i indicates the value of entry i in any solution.

Finally, if no conflict was detected, we have to determine x_i for $i \in [n]$ with $L^i = -1$. Similarly to the procedure in $\mathcal{A}$, we check the effect of fixing such an entry to 1 w.r.t. the objective function. But now, we have to apply the fixings for each P_γ, $\gamma \in \Gamma^\star$. Hence, determining the remaining entries can be performed in $\mathcal{O}(|\Gamma|\, n)$ time. Thus, if we fix a critical row distribution κ, we can solve the corresponding optimization problem in $\mathcal{O}(|\Gamma|\, n)$ time. By iterating over all critical row distributions $\kappa \in [n+1]^{\Gamma^\star}$, we can find the optimal solution of the optimization problem over $\mathrm{S}(\Gamma)$. For this reason, the optimization complexity is in $\mathcal{O}(|\Gamma|\, n^{|\Gamma|})$. $\square$

Corollary 26 *The extension complexity of* $\mathrm{S}(\Gamma)$ *is in* $\mathcal{O}(n^{|\Gamma|})$ *for any* $\Gamma \leq \mathcal{S}_n$. *In particular, the optimization and extension complexity of* $\mathrm{S}(\Gamma)$ *is polynomial, if* $|\Gamma|$ *is constant.*

Proof The bound on the extension complexity can be shown similarly as in the proof of Proposition 24. $\square$

Consequently, if we are given a group which permutes the columns of binary $(m \times n)$-matrices with a fixed number of columns n according to a group $\Gamma \leq \mathcal{S}_n$, such that the columns are ordered lexicographically, we can optimize in polynomial time over $\mathrm{S}(\Gamma)$. These symretopes are full orbitopes w.r.t. Γ, cf. [32].

3.2 IP-formulations for symretopes via symresacks

In Sect. 2 we have discussed the applicability of symretopes and their facets in symmetry handling of binary programs. Since we cannot expect that a "nice" complete linear description of symretopes is always available, cf. Proposition 2, we are interested in an IP-formulation of symretopes, i.e., a set of inequalities that enforce a binary vector to be contained in $\mathrm{S}(\Gamma)$. Adding an IP-formulation of $\mathrm{S}(\Gamma)$ to a binary program allows to break all the symmetry of Γ. Thus, one can benefit from the symmetry breaking effect of $\mathrm{S}(\Gamma)$ without knowing a complete linear description of the symretope.

A possible IP-formulation is given by the FD-inequalities for all permutations in Γ, since these were used in the definition of symretopes. But, as mentioned above, from a computational point of view this may not be favorable, since the coefficients of an FD-inequality can grow exponentially large. For this reason, the aim of this section is to derive an IP-formulation for symretopes in which all coefficients of the constraint matrix are contained in $\{0, \pm 1\}$. Our approach exploits the concept of minimal cover inequalities of knapsack polytopes, and we will demonstrate that the derived formulation is computationally effective in Sect. 4.

Given a binary knapsack polytope $\tilde{P} = \mathrm{conv}(\{x \in \{0,1\}^n : \alpha^\top x \leq \beta\})$, where $\alpha \in \mathbb{R}^n_{\geq 0}$ and $\beta > 0$, a *cover* of $\tilde{P}$ is a set $C \subseteq [n]$ such that $\alpha^\top \chi^C > \beta$. A cover C is called *minimal*, if no proper subset of C is a cover of $\tilde{P}$. In this case, the *minimal cover inequality* is

$$\sum_{i \in C} x_i \leq |C| - 1, \tag{12}$$

which is valid for $\tilde{P}$. It enforces that not all variables in the cover can simultaneously attain value 1. A well-known result, see, e.g., Balas and Jeroslow [6], is that

$$\tilde{P} = \mathrm{conv}(\{x \in \{0,1\}^n : x \text{ fulfills (12) for all minimal covers C of } \tilde{P}\}).$$

Thus, we can replace the knapsack inequality $\alpha^\top x \leq \beta$ by all minimal cover inequalities to obtain an IP-formulation of $\tilde{P}$ whose inequalities have coefficients only in $\{0, 1\}$ on the left-hand side.

Since the coefficients of an FD-inequality are positive and negative, we cannot apply the concept of minimal cover inequalities directly to symresacks. But by complementing the variables x_i with a non-positive coefficient to $1 - w_i$ and defining $w_i = x_i$ for all remaining indices $i \in [n]$, we can transform the symresack into a standard knapsack polytope with inequality

$$\sum_{i=1}^{n} \alpha_i w_i \leq \beta \coloneqq \sum_{\substack{i \in [n]: \\ i \leq \gamma(i)}} \alpha_i = \sum_{\substack{i \in [n]: \\ i > \gamma(i)}} \alpha_i, \tag{13}$$

where

$$\alpha_i \coloneqq \begin{cases} 2^{n-i} - 2^{n-\gamma(i)}, & \text{if } i \leq \gamma(i), \\ 2^{n-\gamma(i)} - 2^{n-i}, & \text{if } i > \gamma(i), \end{cases}$$

for all $i \in [n]$. We denote the transformed knapsack polytope by $\tilde{\mathrm{P}}_\gamma$. By complementing the variables in minimal cover inequalities for $\tilde{\mathrm{P}}_\gamma$, we obtain the desired IP-formulation of P_γ, in which all coefficients of the constraint matrix are contained in $\{0, \pm 1\}$.

Unfortunately, the number of (transformed) minimal cover inequalities can be exponential in n, and thus, this IP-formulation is only useful if its separation problem can be solved efficiently. To show this, we need the following result.

Lemma 27 *Let $\gamma \in \mathcal{S}_n$ and assume that a linear objective can be maximized over $\{x \in \{0, 1\}^n : x \succ \gamma(x)\}$ in $\mathcal{O}(f(n))$ time for some $f : \mathbb{N} \to \mathbb{R}$. Then the separation problem of (transformed) minimal cover inequalities for P_γ with respect to a vector $\bar{x} \in [0, 1]^n$ can be solved in $\mathcal{O}(\max\{n, f(n)\})$ time.*

Proof Let $\bar{x} \in [0, 1]^n$. To be able to separate $\bar{x}$ by (transformed) minimal cover inequalities of P_γ, we complement the entries of $\bar{x}$ whose coefficients in the FD-inequality for γ are non-positive. This yields a vector $\bar{y} \in [0, 1]^n$ in the space of $\tilde{\mathrm{P}}_\gamma$. To solve the separation problem of minimal cover inequalities for $\tilde{\mathrm{P}}_\gamma$ w.r.t. $\bar{y}$, we use the classical approach by Crowder et al. [11], which formulates the separation problem as the integer program

$$\max \left\{ \sum_{i=1}^{n} \bar{y}_i \, y_i - \left(\sum_{i=1}^{n} y_i - 1 \right) : \alpha^\top y \geq \beta + 1, \ y \in \{0, 1\}^n \right\}. \tag{14}$$

If the optimal objective value is positive, y is the incidence vector of a minimal cover for $\tilde{\mathrm{P}}_\gamma$ whose corresponding cover inequality is violated by $\bar{y}$. Otherwise, we have proven that no violated cover inequality exists.

By complementing the variables again, the objective function can be written as $w^\top x$ for some $w \in \mathbb{Z}^n$, and the separation problem (14) turns into

$$\begin{aligned} &\max \{w^\top x \,:\, \bar{c}^\top \gamma(x) \geq \bar{c}^\top x + 1,\ x \in \{0,1\}^n\} \\ &\quad = \max \{w^\top (\mathbb{1} - \tilde{x}) \,:\, \bar{c}^\top (\mathbb{1} - \gamma(\tilde{x})) \geq \bar{c}^\top (\mathbb{1} - \tilde{x}) + 1,\ \tilde{x} \in \{0,1\}^n\} \\ &\quad = \max \{w^\top \mathbb{1} - w^\top \tilde{x} \,:\, \bar{c}^\top \gamma(\tilde{x}) \leq \bar{c}^\top \tilde{x} - 1,\ \tilde{x} \in \{0,1\}^n\} \\ &\quad = \max \{w^\top \mathbb{1} - w^\top \tilde{x} \,:\, \tilde{x} \succ \gamma(\tilde{x}),\ \tilde{x} \in \{0,1\}^n\}. \end{aligned}$$

The optimal solution $\tilde{x}$ of this maximization problem can be found in $\mathcal{O}(f(n))$ time by assumption and can be transformed in linear time into a violated (transformed) minimal cover inequality for P_γ (if it exists). □

Theorem 28 *Given a point $\bar{x} \in [0,1]^n$, the separation problem of minimal cover inequalities for the symresack P_γ, $\gamma \in \mathcal{S}_n$, can be solved in $\mathcal{O}(n\,\alpha(n))$ time.*

Proof By Lemma 27, the separation complexity of minimal cover inequalities for P_γ is bounded by the complexity of linear optimization over $\{x \in \{0,1\}^n \,:\, x \succ \gamma(x)\}$. This optimization problem is very similar to the optimization problem over P_γ. The only difference is that we have to ensure that we use a critical row, i.e., $c \neq n+1$, in the optimization algorithm over Q_γ, see Theorem 23.

Hence, the separation problem of minimal cover inequalities for P_γ can be solved in $\mathcal{O}(n\,\alpha(n))$ time due to Theorem 23 and Lemma 27. □

Theorem 28 allows to obtain a tractable IP-formulation as follows. Since the symretope $\mathrm{S}(\Gamma)$ can be equivalently defined as in (11), an IP-formulation for the maximization problem of $c \in \mathbb{R}^n$ over $\mathrm{S}(\Gamma)$ is given by

$$\max \left\{c^\top x \,:\, x \in \{0,1\}^n \text{ satisfies all min. cover ineq. of } \mathrm{P}_\gamma,\, \gamma \in \Gamma\right\}. \tag{15}$$

By Theorem 28, the IP-formulation (15) can be separated in time $\mathcal{O}(|\Gamma|\, n\,\alpha(n))$. Hence, if $|\Gamma| \in \mathcal{O}(n^k)$ for some $k \geq 0$, the separation problem for (15) can be solved in polynomial time.

Moreover, observe that the FD-inequalities for a group Γ can be separated in $\mathcal{O}(|\Gamma|\, n)$ time. Thus, the worst-case run time increases only by a sublinear factor if we use the IP-formulation (15) instead of FD-inequalities, but we gain numerical stability.

3.3 Minimal cover separation for orbisacks

Full orbitopes w.r.t. $\mathcal{S}_n$, see Sect. 2.2.2, are important since they appear in many applications, see Sect. 4. For this reason, we will focus on orbitope actions in this section in more detail. First, we show that it suffices to consider $n-1$ orbisacks to obtain an IP-formulation of full orbitopes. Afterwards, we prove that the separation problem of minimal cover inequalities for orbisacks can be solved in linear time. Thus, symmetry in many applications can be handled efficiently.

Recall from Sect. 2.2.2 that an orbitope action of $\gamma \in \mathcal{S}_n$ on $(m \times n)$-matrices X is to reorder the columns according to γ. Thus, the corresponding symmetry group is a subgroup of $\mathcal{S}_{[m]\times[n]}$ that is isomorphic to $\mathcal{S}_n$. Moreover, we use the convention of the universal ordering vector for matrices introduced in Sect. 2.2.3. Furthermore, we denote the group that is generated by τ with $\langle\tau\rangle$.

Proposition 29 *Let $\Gamma \leq \mathcal{S}_{[m]\times[n]}$ be the group that permutes the columns of binary $(m \times n)$-matrices arbitrarily, i.e., $\Gamma \simeq \mathcal{S}_n$, and denote by τ_i, $i \in [n-1]$, the permutation which exchanges column i and $i+1$ of such a matrix. Then, an IP-formulation of $\mathrm{S}(\Gamma)$ is given by $\bigcap_{i=1}^{n-1} \mathrm{S}(\langle\tau_i\rangle)$.*

Proof By the definition of the ordering vector, the symretope $\mathrm{S}(\Gamma)$ is the full orbitope w.r.t. $\mathcal{S}_n$. Thus, its vertices are binary $(m \times n)$-matrices whose columns are sorted lexicographically non-increasing. Moreover, τ_i, $i \in [n-1]$, exchanges column i and $i+1$ of any such matrix, and thus, is the orbitope O_m on these two columns. Hence,

$$\mathrm{S}(\langle\tau_i\rangle) = [0,1]^{m\times(i-1)} \times \mathrm{O}_m \times [0,1]^{m\times(n-(i+1))}$$

by Proposition 5. Hence, the vertices of $\mathrm{S}(\langle\tau_i\rangle)$ are all those binary matrices whose i-th column is lexicographically not smaller than its $(i+1)$-st column. Consequently, the intersection $S^\star = \bigcap_{i=1}^{n-1} \mathrm{S}(\langle\tau_i\rangle)$ contains exactly those binary matrices whose columns are sorted lexicographically non-increasing. Thus, $S^\star$ is an IP-formulation of $\mathrm{S}(\Gamma)$. □

Observe that $\langle\tau_i\rangle$ contains only one non-trivial permutation, namely τ_i itself. Hence, restricting the IP-formulation (15) of the full orbitope w.r.t. $\mathcal{S}_n$ to the permutations τ_i, $i \in [n-1]$, instead of all permutations in $\mathcal{S}_n$, suffices to handle the symmetries of a full orbitope *completely* and it can be separated in $\mathcal{O}(m\, n\, \alpha(m))$ time by Theorem 28. In the remainder of this section, we show that we can improve the separation complexity to $\mathcal{O}(mn)$.

Proposition 30 *Given a point $\bar{x} \in [0,1]^{m\times 2}$, the separation problem of minimal cover inequalities for the orbisack O_m can be solved in $\mathcal{O}(m)$ time.*

Proof Since the orbisack $\mathrm{O}_m \subseteq \{0,1\}^{m\times 2}$ is the symresack for the permutation τ that exchanges both columns of an $(m \times 2)$-matrix, an IP-formulation of the orbisack is given via the FD-inequality for τ and binary constraints. Analogously to Theorem 28, we can separate minimal cover inequalities of the orbisack by solving a linear optimization problem over $\{x \in \{0,1\}^n : x \succ \tau(x)\}$. By a minor modification of the optimization algorithm presented by Loos [37, Thm. 3.33] (again, we have to ensure that we use a critical row), this problem can be solved in $\mathcal{O}(m)$ time by Lemma 27. □

Corollary 31 *An IP-formulation of the full orbitope w.r.t. $\mathcal{S}_n$ is given by restricting (15) to the permutations τ_i, $i \in [n-1]$ and binary constraints. This IP-formulation can be separated in $\mathcal{O}(mn)$ time.*

3.4 Polyhedral properties of symresacks

The aim of this section is to provide basic polyhedral properties of symresacks. In particular, we will link some of the polyhedral results on symresacks to symretopes, which shows that it can be worthwhile to analyze the facial structure of symresacks to get a better understanding of symretopes.

Recall that a symresack is a non-standard knapsack polytope because the coefficients of an FD-inequality are positive and negative. Hence, we cannot apply classical results for knapsack polytopes directly to symresacks. But by complementing the variables with a non-positive coefficient as in (13), we obtain the transformed symresack $\tilde{\mathrm{P}}_\gamma$ which is a knapsack polytope.

Lemma 32 *Let $\gamma \in \mathcal{S}_n$. Then,* $\dim(\mathrm{P}_\gamma) = n$*, i.e.,* P_γ *is full-dimensional.*

Proof By definition, we have that $\mathrm{S}(\mathcal{S}_n) \subseteq \mathrm{P}_\gamma$. Consequently, P_γ is full-dimensional by Lemma 3. □

Lemma 7 gives sufficient conditions for the trivial inequalities to define facets of symretopes. For the special case of symresacks a complete characterization is as follows.

Proposition 33 *Let $\gamma \in \mathcal{S}_n$. The inequality $x_i \geq 0$, $i \in [n]$, defines a facet of* P_γ *if and only if*

- *$i \geq \gamma(i)$ or*
- *$i < \gamma(i)$ and $\alpha_i + \alpha_j \leq \beta$ for all $j \in [n]$ with $j > \gamma(j)$.*

Furthermore, $x_i \leq 1$, $i \in [n]$, defines a facet of P_γ *if and only if*

- *$i \leq \gamma(i)$ or*
- *$i > \gamma(i)$ and $\alpha_i + \alpha_j \leq \beta$ for all $j \in [n]$ with $j < \gamma(j)$.*

Proof Since the transformed symresack $\tilde{\mathrm{P}}_\gamma$ is a full-dimensional knapsack polytope induced by an inequality with non-negative coefficients, it is a down-monotone polytope. Hence, the inequalities $w_i \geq 0$, $i \in [n]$, define facets of $\tilde{\mathrm{P}}_\gamma$, see Hammer et al. [19]. Observe that $w_i \geq 0$ transforms to $x_i \geq 0$ if $i \geq \gamma(i)$ and to $x_i \leq 1$ if $i < \gamma(i)$. Consequently, $x_i \geq 0$ defines a facet of P_γ if $i \geq \gamma(i)$, and $x_i \leq 1$ defines a facet of P_γ if $i < \gamma(i)$. Furthermore, $x_i \geq 0$ and $x_i \leq 1$ define facets for P_γ if $i = \gamma(i)$, because the coefficient of i in the FD-inequality is 0.

Let $i \in [n]$ with $i < \gamma(i)$. The inequality $x_i \geq 0$ defines a facet of P_γ if and only if $w_i \leq 1$ defines a facet of $\tilde{\mathrm{P}}_\gamma$. Following Balas [4], $w_i \leq 1$ defines a facet of $\tilde{\mathrm{P}}_\gamma$ if and only if $\alpha_i + \alpha^\star \leq \beta$, where

$$\alpha^\star := \max\{\alpha_j \ : \ j \in [n]\setminus\{i\}\}.$$

By construction of β, we have $\alpha_i + \alpha_j \leq \beta$ for each $j \in [n]\setminus\{i\}$ with $j \leq \gamma(j)$. Thus, it suffices to consider in the definition of $\alpha^\star$ only the variables $j \in [n]$ with $j > \gamma(j)$, and the assertion follows.

The second case can be shown analogously. □

From Sect. 2 recall the partial characterization when trivial constraints $x_i \le 1$ or $x_i \ge 0$ define facets of symretopes. Due to Proposition 33, we are now able to extend this characterization by a necessary condition for a trivial constraint to define a facet of $\mathrm{S}(\Gamma)$.

Corollary 34 *Let $\Gamma \le \mathcal{S}_n$. If a trivial constraint for variable x_i, $i \in [n]$, defines a facet of $\mathrm{S}(\Gamma)$, then it defines a facet of P_γ for all $\gamma \in \Gamma$.*

Proof Since $\mathrm{S}(\Gamma)$ and P_γ, $\gamma \in \Gamma$, are 0/1-polytopes, each vertex of $\mathrm{S}(\Gamma)$ is a vertex of P_γ for every $\gamma \in \Gamma$. Thus, if a trivial constraint defines a facet of $\mathrm{S}(\Gamma)$, there are n affinely independent vertices of $\mathrm{S}(\Gamma)$ that fulfill the constraint with equality, since $\mathrm{S}(\Gamma)$ is full-dimensional. Consequently, these n vertices are contained in every symresack P_γ, and thus, the trivial constraint defines a facet of each of those symresacks due to Lemma 32. □

In Sect. 2.1 we studied cases in which FD-inequalities define facets of symretopes, see Proposition 11. However, a complete characterization is not known. For symresacks, such a characterization is as follows:

Proposition 35 *Let $\gamma \in \mathcal{S}_n$. The FD-inequality for γ defines a facet of P_γ if and only if γ is a transposition.*

Proof If $\gamma = \mathrm{id}_{\mathcal{S}_n}$, the FD-inequality is $0 \ge 0$, and thus, does not define a facet of P_γ. For this reason, we can assume in the following that $\gamma \neq \mathrm{id}_{\mathcal{S}_n}$.

Let $\zeta_1 \circ \cdots \circ \zeta_m$ be the disjoint cycle decomposition of γ, where we include the trivial cycles (j) for each fixed point $j \in [n]$ of γ. Moreover, let F be the face of P_γ that is induced by the FD-inequality.

To prove that the FD-inequality for γ does not define a facet of P_γ if γ is not a transposition, we exploit the following claim.

Claim Let $N \subseteq [n]$ be such that $\bar{c}^\top \chi^N = \bar{c}^\top \gamma(\chi^N)$. Then $N = \bigcup_{i \in A} \mathrm{supp}(\zeta_i)$ for some $A \subseteq [m]$. In particular, if $N = \mathrm{supp}(\zeta_j)$ for some $j \in [m]$, then $\bar{c}^\top \chi^N = \bar{c}^\top \gamma(\chi^N)$.

Proof (of claim) If $\bar{c}^\top \chi^N = \bar{c}^\top \gamma(\chi^N)$, then $N = \gamma(N)$ and thus, γ stabilizes N. Consequently, $N = \bigcup_{i \in A} \mathrm{supp}(\zeta_i)$ for some $A \subseteq [m]$, because each $i \in N$ is contained in exactly one cycle ζ_j and the only subsets of $\mathrm{supp}(\zeta_j)$ that are stabilized by ζ_j are $\varnothing$ and $\mathrm{supp}(\zeta_j)$. □

Let $\bar{x} \in \{0, 1\}^n$ be such that $\bar{c}^\top \bar{x} = \bar{c}^\top \gamma(\bar{x})$ and $N = \mathrm{supp}(\bar{x})$, i.e., $\bar{x} = \chi^N$. By the claim, we have that $N = \bigcup_{i \in A} \mathrm{supp}(\zeta_i)$ for some $A \subseteq [m]$, i.e., $\bar{x}$ is constant along $\mathrm{supp}(\zeta_i)$. Hence, if there is a cycle ζ_i of length at least three, the claim proves that there are at most $n - |\mathrm{supp}(\zeta_i)|$ degrees of freedom outside $\mathrm{supp}(\zeta_j)$, while on $\mathrm{supp}(\zeta_j)$ there is exactly one degree of freedom for any vertex of P_γ. For this reason, there can be at most $n - \mathrm{supp}(\zeta_j) + 1 \le n - 2$ linearly independent vertices of P_γ in F. Consequently, the FD-inequality cannot define a facet of P_γ.

Conversely, if each cycle ζ_i has a length of at most two, we proceed as follows: Let k be the number of cycles ζ_i of length 2. Then, γ has $n - 2k$ fixed points. Similarly as above, we have on each cycle ζ_i exactly one degree of freedom in any vertex of P_γ in F. Since $0 \in F$, there are $(n-2k)+k = n-k$ linearly independent vertices of P_γ in F, and thus, there are $n-k+1$ affinely independent vertices of P_γ in F. For this reason, F defines a facet of P_γ if and only if $k = 1$, i.e., γ is a transposition. $\square$

Corollary 36 *The polytope P_γ, $\gamma \in \mathcal{S}_n^\star$, is completely described by the trivial inequalities and the FD-inequality for γ if and only if γ is a transposition or the identity.*

Proof Necessity follows by Proposition 35. To prove sufficiency, we observe that the FD-inequality of the transposition (i, j), $i < j$, is $-x_i + x_j \leq 0$. Hence, the constraint matrix of the FD-inequality and trivial constraints is totally unimodular, and thus, the corresponding polytope is integral. $\square$

Furthermore, we can exploit knowledge on the facial structure of P_γ to get the facial structure of $\mathrm{P}_{\gamma^{-1}}$.

Proposition 37 *Let $\gamma \in \mathcal{S}_n$. Then P_γ and $\mathrm{P}_{\gamma^{-1}}$ are linearly equivalent.*

Proof To prove this proposition, it suffices to show that $\tilde{\mathrm{P}}_\gamma$ and $\tilde{\mathrm{P}}_{\gamma^{-1}}$ are linearly equivalent. Observe that the coefficient of variable $i \in [n]$ in the FD-inequality of γ coincides with the negative value of the coefficient of $\gamma(i)$ in the FD-inequality for γ^{-1}:

$$2^{n-i} - 2^{n-\gamma(i)} = -\left(2^{n-\gamma(i)} - 2^{n-i}\right) = -\left(2^{n-\gamma(i)} - 2^{n-\gamma^{-1}(\gamma(i))}\right).$$

Thus, we can permute the knapsack inequality of $\tilde{\mathrm{P}}_\gamma$ onto the knapsack inequality of $\tilde{\mathrm{P}}_{\gamma^{-1}}$. Hence, $\tilde{\mathrm{P}}_\gamma$ and $\tilde{\mathrm{P}}_{\gamma^{-1}}$ are isomorphic, and the assertion follows. $\square$

The previous results of this section provide basic knowledge on symresacks, and these results enable us also to get a better understanding of symretopes $\mathrm{S}(\varGamma)$ by investigating the symresacks P_γ for all $\gamma \in \varGamma$. In particular, facet defining inequalities of symresacks are natural candidates to be used as cutting planes for symretopes, and thus, as symmetry handling inequalities in binary programs. But observe that facet inequalities of symresacks need, in general, not define facets of symretopes. For the important example of full orbitopes, however, facet defining inequalities of particular orbisacks can be trivially lifted to facets of the orbitope.

Proposition 38 *Let $\varGamma \leq \mathcal{S}_{[m]\times[n]}$ be the group that permutes the columns of binary $(m \times n)$-matrices arbitrarily, in particular, $\varGamma \simeq \mathcal{S}_n$, and denote by τ_j, $j \in [n-1]$, the permutation that exchanges column j and $j+1$ of such a matrix. Then, any non-trivial facet defining inequality of $\mathrm{P}_{\tau_j} = \mathrm{S}(\langle \tau_j \rangle)$ also defines a facet of $\mathrm{S}(\varGamma)$.*

Proof Recall from Sect. 3.3 that $\mathrm{S}(\langle \tau_j \rangle) = [0, 1]^{m\times(i-1)} \times \mathrm{O}_m \times [0, 1]^{m\times(n-(i+1))}$. Consequently, any non-trivial facet defining inequality of $\mathrm{S}(\langle \tau_j \rangle)$ contains only variables of columns j and $j+1$, and thus, is equivalent to a facet defining inequality of the orbisack O_m. To show that it defines a facet of $\mathrm{S}(\varGamma)$, we use the following claim.

Claim Every non-trivial facet F of O_m contains m vertices of O_m whose first columns are affinely independent and are not equal to $\mathbb{1}$. Moreover, F contains m vertices of O_m whose second columns are affinely independent and are not equal to 0.

Proof of (claim) See "Appendix B".

We are now able to show that every non-trivial facet F of O_m trivially lifts to a facet of $\mathrm{S}(\Gamma)$ by constructing mn affinely independent vertices of $\mathrm{S}(\Gamma)$ that fulfill the facet defining inequality of O_m with equality.

First, we know that there exist $2m$ affinely independent vertices X_i of F, since F is a facet of O_m. Each such vertex X_i can be extended to a vertex

$$\lambda(X_i) \coloneqq (\mathbb{1}_{m\times(j-1)}, X_i, 0_{m\times(n-(j+1))}) \in \{0,1\}^{m\times n}$$

of $\mathrm{S}(\Gamma)$ that fulfills the trivially lifted facet defining inequality with equality. Here, the subscripts of $\mathbb{1}$ and 0 are the dimension of the all-1 matrix and all-0 matrix.

Second, we construct the missing $mn - 2m$ vertices that fulfill the inequality with equality. By the claim there exist m vertices $\bar{X}_i$, $i \in [m]$, of F whose first columns are affinely independent and which are not $\mathbb{1}$. Let $\bar{X}_i^k$ be the matrix that is obtained by replacing columns $j-k, \ldots, j-1$ of $\lambda(\bar{X}_i)$ by the first column of $\bar{X}_i$. Moreover, the claim guarantees the existence of m vertices $\underline{X}_i$, $i \in [m]$, of F whose second columns are affinely independent and that are not 0. Let $\underline{X}_i^k$ be the matrix that is obtained by replacing columns $j+2, \ldots, j+1+k$ of $\lambda(\underline{X}_i)$ by the second column of $\underline{X}_i$.

Each of the newly constructed matrices is a vertex of $\mathrm{S}(\Gamma)$, because their columns are sorted lexicographically non-increasing. Moreover, one can show that the matrices

$$\begin{aligned}&\{\bar{X}_i^k \,:\, (k,i) \in [j-1]\times[m]\} \cup \{\lambda(X_i) \,:\, i \in [2m]\}\\ &\quad\cup \{\underline{X}_i^k \,:\, (k,i) \in [n-(i+1)]\times[m]\}\end{aligned}$$

are affinely independent. Consequently, the trivially lifted facet defining inequality defines a facet of $\mathrm{S}(\Gamma)$. ◻

Remark 39 Thus, the facets for orbisacks of Kaibel and Loos [30] yield trivially lifted facets for the corresponding symretopes. In particular, some of these facets are defined by minimal cover inequalities into which at most one variable is lifted, see [25,30].

Finally, we present some examples for symresacks w.r.t. specific cyclic shifts σ_s by s positions, i.e., $\sigma_s(i) = ((i+s-1) \bmod n) + 1$ for $s \in [n]$.

Lemma 40 *The polytope* P_{σ_1} *is completely described by*

$$-x_i + x_n \le 0, \quad \forall\, i \in [n-1], \tag{16}$$

$$x_i \in [0,1], \quad \forall\, i \in [n]. \tag{17}$$

Proof Because the inequalities (16) induce the incidence matrix of a directed graph, the induced matrix is totally unimodular, and the polytope which is defined by (16) and (17) is integral. Hence, it suffices to prove that this is an IP-formulation for P_{σ_1}.

Let $x \in \{0,1\}^n$. If $x_n = 0$, the shifted vector $\sigma_1(x)$ is obviously lexicographically not greater than x, and x fulfills all of the given inequalities. Conversely, if $x \neq \mathbb{1}$ is a binary vector with $x_n = 1$, denote with i the smallest index such that $x_i = 0$. Then the first i positions of $\sigma_1(x)$ are 1-entries, which proves that $\sigma_1(x) \succ x$. Hence, $x = \mathbb{1}$ is the only binary vector with $x_n = 1$ which is contained in P_{σ_1}. This is enforced by (16). Thus, the binary points that fulfill all constraints of the proposed inequality system are exactly the vertices of P_{σ_1} by distinguishing whether $x_n = 0$ or $x_n = 1$. $\square$

By Proposition 37, we get the following result for $\mathrm{P}_{\sigma_{n-1}}$:

Corollary 41 *The polytope* $\mathrm{P}_{\sigma_{n-1}}$ *is completely described by*

$$\begin{aligned} -x_1 + x_i &\leq 0, && \forall\, i \in \{2, \dots, n\}, \\ x_i &\in [0, 1], && \forall\, i \in [n]. \end{aligned}$$

If n is even and we consider the cyclic shift by $m = \frac{n}{2}$ positions, the corresponding symresack is isomorphic to the orbisack O_m. But for arbitrary cyclic shifts, the symresacks are much more complicated. In particular, complete linear descriptions for further cyclic shifts are currently unknown.

4 Computational experience

In Sect. 3.2 we developed an IP-formulation for symretopes which is based on symresacks and uses only inequalities with coefficients in $\{0, \pm 1\}$ of the normal vector. To investigate the impact of this formulation on the solution process of symmetric (mixed) binary programs within a branch-and-cut-and-propagate algorithm, we implemented plugins for `SCIP` [1,38] that contain separation and propagation routines for symresacks and for orbisacks. Both implementations, which are written in `C`, will be described in more detail below. Furthermore, we have written a propagation routine for full orbitopes.

We embedded both plugins in the symmetry handling code that is described in [47]; the extended code is publicly available on the second author's web page. Briefly summarized, the implementation of [47] allows to compute a symmetry group of a mixed integer program and provides several symmetry handling techniques like isomorphism pruning by Margot [39] and orbital branching by Ostrowski et al. [46], but also some basic symmetry handling inequalities, e.g., the inequalities $-x_i + x_{i+1} \leq 0$, $i \in [n-1]$, for $\mathcal{S}_n$, cf. Example 15.

Furthermore, all results of our experiments are provided in an online supplement which is provided on the second author's web page.

4.1 Implementation details

The symresack plugin provides an implementation of the separation procedure of minimal cover inequalities derived in Sect. 3.2. Furthermore, the FD-inequality for a permutation γ is propagated by iterating over $i = 1, \dots, n$ and applying the following operations in each step:

- Fixing the variable $x_i = 1$ if $x_{\gamma^{-1}(i)} = 1$, or $x_{\gamma^{-1}(i)} = 0$ if $x_i = 0$,
- pruning a node if $x_i = 0$ and $x_{\gamma^{-1}(i)} = 1$,
- terminating if x_i and $x_{\gamma^{-1}(i)}$ have not been fixed yet or if $x_i = 1$ or $x_{\gamma^{-1}(i)} = 0$.

In the last step, no propagation for succeeding indices is applicable since the lexicographic order depends on the entries of a vector in increasing order. Both the separation and propagation procedure are called for all nodes in each fifth level of the branch-and-bound tree. Finally, the symresack plugin contains a routine that checks whether the corresponding permutation is a composition of disjoint 2-cycles. In this case, the symresack is an orbisack and we treat it by the orbisack plugin, described next.

The orbisack plugin provides a similar propagation method as the symresack plugin and a separation routine for minimal cover inequalities. This routine separates minimal cover inequalities of orbisacks using the algorithm in the proof of Proposition 30. As for symresacks, we call the propagation and separation routines at each fifth level of the branch-and-bound tree. Observe that an efficient algorithm to separate facets of orbisacks is provided by Loos [37]. However, preliminary experiments indicate that we cannot benefit from adding facets instead of minimal covers. As a consequence, we decided to only add minimal cover inequalities, since their coefficients are in $\{0, \pm 1\}$.

Moreover, we add for each symmetry that is handled by an orbisack or symresack constraint the initial inequality

$$x_1 \geq x_{\gamma^{-1}(1)} \tag{18}$$

to the problem.

Furthermore, we implemented two heuristics to detect orbitope actions, since these actions may not be detected by the symmetry code of [47] if they are induced by a subgroup of the symmetry group of the binary program. Because an orbitope action is generated by the action of some orbisacks, cf. Proposition 29, we select all generators of the symmetry group which are compositions of 2-cycles. We keep only those generators which contain the maximal or minimal number of 2-cycles and heuristically try to combine a subset of these generators to an orbitope action.

Example 42 Consider the symmetry group generated by $(1, 2)(3, 4)(5, 6)$, $(1, 3)(2, 4)$, and $(3, 5)(4, 6)$. Using the min-heuristic detects a (2×3)-orbitope according to the following variable ordering

$$\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}.$$

Applying the max-heuristic, however, finds a (3×2)-orbitope for the variable ordering

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}.$$

Thus, it is possible that both heuristics detect different orbitopes. By rearranging the universal ordering vector, one of both orbitopes can be used to handle symmetries in a binary program, but each of the corresponding groups forms a proper subgroup of the symmetry group.

Finally, our propagation routine for full orbitopes (w.r.t. $\mathcal{S}_n$) exploits the vertex structure of full orbitopes described by Loos [37, Lem. 3.13]. We iterate over the entries of row $i = 1$ of a (partial) solution $X \in \mathbb{R}^{m \times n}$ for the orbitope and

- fix all entries that are to the left of the right-most 1-entry j_1 in row i to 1, and
- fix all entries that are to the right of the left-most 0-entry j_0 in row i to 0.

If either of these fixings produces a contradiction to already fixed entries of X, we can prune the current node. Otherwise, we increase i by one and we call the propagation routine recursively for the suborbitopes of the first j_1 columns and the last j_0 columns.

4.2 Symmetry handling settings

In our experiments, we tested three classes of settings. To be able to find orbitopes heuristically, in the setting `orbi-max` and `orbi-min`, we activate the heuristic that uses the maximal and minimal number of 2-cycles in a permutation, respectively, cf. Sect. 4.1. The handling of symmetries via symresacks is deactivated in both these settings. Conversely, we allow symresacks in the third setting `symre`, but both orbitope heuristics are disabled. In "Appendix D", we describe in detail which symmetry handling inequalities are added by these three settings.

Additionally, we wanted to check the impact of the separation and propagation routines of the aforementioned plugins on the performance of the solver. For this reason, we ran our experiments with three variants of the settings `symre`, `orbi-max`, and `orbi-min`: either the separation (`s`), propagation (`p`), or both routines (`sp`) are active. Furthermore, we compared our techniques with two further methods, which are known in the literature and that are available in the implementation of [47]: `ISP-NST` is a variant of isomorphism pruning, see Margot [39,40], in which symmetry handling in a subtree of a solution in the branch-and-bound tree is deactivated if the stabilizer group of this solution is trivial. For each factor $\varGamma'$ of $\varGamma$, `S-orbitmin` searches for the smallest index i that is affected by $\varGamma'$ and adds the following inequalities to the problem during presolving:

$$x_i \geq x_k \tag{19}$$

for all $k \in \varGamma'(i) \setminus \{i\}$, see also Liberti [35] and Remark 9. These inequalities handle some symmetries within the orbit $\varGamma'(i)$ and proved to be relatively effective.

4.3 Numerical results for highly symmetric instances

To check the impact of our symmetry handling techniques on the solution process, we applied our code on two types of instances: highly symmetric instances, which will be described in this section, and general benchmark instances, see Sect. 4.4. The experiments on both test sets were performed with the symmetry code of [47] and the previously described plugins using a preliminary version of `SCIP 4.0.1`, see Achterberg [1] and [38], and `CPLEX 12.6.1` [26] as LP-solver. The tests were run on a Linux cluster with Intel E5 3.5 GHz quad core processors and 32 GB memory; the code was run using one thread and running a single process at a time. The time limit was set to 3600 s per instance.

Our test set of highly symmetric instances is formed by instances of two different problems that were discussed in the literature:

- The aim of the *SONET Network Design Problem (SONET)* is to design a cost minimal network of SONET rings as well as to find a routing strategy of signals through this network, see Sherali and Smith [51]. The symmetry groups of SONET instances lead to partitioning orbitopes.
- The *Wagon Load-Balancing Problem (WB)* is to assign boxes of different weights to wagons such that each wagon receives the same number of boxes, minimizing the total absolute deviation of wagon loads from the average weight of a wagon, see Ghoniem and Sherali [17]. Here, we can permute the wagons and boxes of identical weight arbitrarily; the symmetry groups of these instances act as for partitioning orbitopes, but with additional row permutations.

Table 1 summarizes the results of our experiments on 50 randomly generated SONET instances and 120 WB instances by Ghoniem and Sherali. We use the *shifted geometric mean*

$$\left(\prod(t_i + s)\right)^{1/n} - s$$

with shift s. We use a shift $s = 10$ for time and $s = 100$ for nodes in order to decrease the strong influence of the very easy instances in the mean values. The column "#nodes" in Table 1 gives the shifted geometric mean of the number of nodes needed during the solution process, whereas column "time" provides the shifted geometric mean of time per instance. In column "#opt" we list the number of instances that could be solved to optimality within the time limit and "#act" shows in how many instances symmetry handling techniques were applied. Finally, columns "method-time" and "sym-time" summarize the shifted geometric mean of the time spent for symmetry handling techniques and computing symmetries.

For the SONET instances, we can observe that handling symmetry by separating minimal cover inequalities of symresacks or orbisacks is most effective (speed up about 96%); only propagating both types of constraint is less effective and leads to a speed up of about 93%. Furthermore, the average solution time per instance for the settings `orbi-max`, `orbi-min`, and `symre` is almost the same, because all settings add the same orbisack constraints to the problem for all instances. `ISP-NST` leads to a comparable speed up (96.2%), whereas `S-orbitmin` reduces the solution time by only 63.5%. Nevertheless, the latter result is remarkable, since the additional constraints of `S-orbitmin` involve only variables of one orbit.

Since all SONET instances can be solved to optimality within 1 h by each setting, this allows for a fair comparison of the number of nodes in the branch-and-bound tree needed by the different settings to solve the problems. Similar as for the solution time, the fewest number of nodes on average for the `symre` and `orbi` settings were needed by the settings that separate minimal cover inequalities; in comparison with the default setting, the number of nodes decreases by up to 98.6%. Although the average solution time of `symre-sp` and `ISP-NST` are comparable, `ISP-NST` needs 33% fewer nodes on average to solve an instance to optimality. Hence, calling `ISP-NST` is more costly per node than any of the `orbi-s` or `symre-s` settings.

Table 1 Comparison of different symmetry handling variants for highly symmetric instances

Setting	#nodes	Time	#opt	#act	Method-time	Sym-time
`SONET` (50)						
default	90890.2	84.96	50	0	0.25	0.00
orbi-max-p	3433.2	6.27	50	50	0.03	0.01
orbi-max-s	1316.4	2.92	50	50	0.01	0.00
orbi-max-sp	1324.7	2.88	50	50	0.02	0.00
orbi-min-p	3433.2	6.29	50	50	0.03	0.00
orbi-min-s	1316.4	2.92	50	50	0.01	0.00
orbi-min-sp	1324.7	2.88	50	50	0.02	0.00
symre-p	3573.6	6.49	50	50	0.03	0.00
symre-s	1306.4	3.22	50	50	0.01	0.01
symre-sp	1250.3	2.96	50	50	0.01	0.00
ISP-NST	829.8	3.19	50	50	0.79	0.00
S-orbitmin	27511.4	31.02	50	50	0.09	0.00
`WB` (120)						
default	19172171.1	3363.88	2	0	16.45	0.00
orbi-max-p	17820067.2	3268.05	2	120	23.32	0.16
orbi-max-s	17503254.4	3265.93	2	120	20.98	0.16
orbi-max-sp	17789094.6	3268.13	2	120	27.90	0.16
orbi-min-p	17652118.6	3266.37	2	120	16.45	0.16
orbi-min-s	17648644.1	3266.34	2	120	16.53	0.16
orbi-min-sp	17665078.8	3266.34	2	120	16.39	0.16
symre-p	17106239.0	3267.76	2	120	24.23	0.16
symre-s	60998.0	51.57	120	120	8.81	0.16
symre-sp	37737.9	32.23	120	120	5.77	0.16
ISP-NST	–	2344.25	18	120	–	–
S-orbitmin	17632656.1	3264.49	2	120	16.29	0.00

Finally, the WB instances turned out to be very hard for the default setting: only 2 of 120 instances could be solved within 1 h. If we use `ISP-NST` to handle symmetry in these instances, we are able to solve 18 instances to optimality; but 55 instances run out of memory while computing stabilizers of solutions. For this reason, Table 1 cannot provide statistics on the number of nodes of the branch-and-bound tree as well as on the time spent to compute symmetries and to perform isomorphism pruning for isomorphism pruning on these instances. However, using `symre-s` or `symre-sp` to handle symmetries allows to solve all 120 instances within about 50 or 30 s, respectively, per instance on average. The remaining settings fail to improve the number of solved instances in comparison to the default setting. A possible explanation for this behavior is that the action of the large and unclassified symmetry groups of these instances is much more complex than orbitope actions. Thus, handling orbitope actions does not suffice to make these instances tractable.

Table 2 Comparison of the geometric mean of closed gap in the root node for different symmetry handling variants

Setting	SONET	WB	Margot1	M2003-sym	M2010-sym	M2010-bench
Gap closed						
default	5.96	1.61	3.88	17.08	9.74	16.83
orbi-max-p	22.08	1.44	2.68	20.47	10.39	18.51
orbi-max-s	21.16	1.52	2.68	21.40	10.44	18.37
orbi-max-sp	22.06	1.44	2.68	21.40	10.44	18.37
orbi-min-p	22.08	1.79	2.68	20.47	10.39	18.51
orbi-min-s	21.16	1.79	2.68	21.40	10.44	18.37
orbi-min-sp	22.06	1.79	2.68	21.40	10.44	18.37
symre-p	16.55	27.86	6.43	20.40	10.52	18.81
symre-s	15.75	33.16	7.28	20.53	10.51	18.82
symre-sp	16.76	33.59	8.59	20.73	10.53	18.84
ISP-NST	5.96	1.98	3.88	17.08	9.74	16.83
S-orbitmin	10.69	1.79	3.40	20.49	10.46	18.31

To further analyze the effect of the symmetry handling constraints, we investigate the dual bounds after solving the root node of the branch-and-bound tree. In Table 2 we compare the the *gap closed* value that is defined as

$$100\% - 100\% \cdot \frac{|p_{\text{best}} - d|}{|p_{\text{best}} - p_{\text{LP}}|}$$

for the different settings, where p_{best} is the best known value of a primal solution, p_{LP} is the value of the (fractional) solution after solving the first linear programming relaxation, and d is the value of the dual bound after solving the root node. This value describes the amount of the primal/dual bound gap closed after solving the root node using cutting planes. Analyzing the WB instances, we can see that the gap closed value for the `symre` settings is one order of magnitude larger than for the remaining settings. Observe that even for `symre-p` the gap closed value is much larger than in the default setting, because we add Inequality (18) for each generator of Γ. Furthermore, note that the gap closed values for the default and `ISP-NST` settings are not 0, because general cutting planes are added during the solving process. A similar observation holds for the SONET instances: if we can benefit from orbisack or symresack constraints, the gap closed value for these settings is significantly larger than for the default setting. This effect is due to further cutting planes (e.g., Gomory inequalities) in combination with the added minimal cover inequalities of symresacks/orbisacks.
Summarizing, we can see that `symre-s` and `symre-sp` are the most effective symmetry handling techniques on the considered highly symmetric instances; either because they allow a significant speed up of the solution process or they allow to solve an instance that cannot be solved by other techniques. Furthermore, the average solution times and gap closed values seem to indicate a positive correlation between

the gap closed value and the effectiveness of these methods. However, a computation of the correlation coefficients of the increase of the gap closed value and the speed up per instance shows that no correlation can be expected, cf. the online supplement of this paper. But if we compute the correlation coefficients of the increase of the gap closed value in comparison with the reduction of nodes in the branch-and-bound tree for instances that can be solved to optimality, we observe a positive correlation.

4.4 Numerical results for benchmark instances

The test set of benchmark instances consists of 16 instances of Margot [40] (Margot1) as well as instances of MIPLIB 2003 [2] (M2003-sym) and MIPLIB 2010 [33], where we used two test sets that were formed by MIPLIB 2010 instances, (M2010-sym) and (M2010-bench). These test sets were also used in [47] and contain (besides M2010-bench) only instances in which symmetries are present.

Table 3 provides a summary of our experiments on these instances. On the Margot instances only 6 of 16 instances are solved to optimality within 1 h by the default setting; if we use an orbisack setting for these instances, we can solve one additional instance and the average solution time improves by about 37.1%. Using a symresack setting to handle symmetry, we can solve up to 13 instances and obtain a speed-up of at most 90.6%, whereas `S-orbitmin` realizes a speed up of 59.3%. ISP-NST, however, is able to solve all but one instance to optimality; the achieved speed up in comparison to the default setting is 97.0%. An explanation for this behavior is that the symmetry groups of these instances are unclassified groups whose actions are much more complex than orbitope actions. Thus, using orbisack settings or `S-orbitmin` can handle only a small amount of the symmetries, whereas the symresack settings or `ISP-NST` can deal with arbitrary permutations. The results for symresacks are remarkable, because many other polyhedral methods, including orbitopes, fail to decrease the solution time or to increase the number of solved instances by this order of magnitude, see [47, Section 5.6] and the results for the `orbi-*` settings.

The situation is not so clear cut on the MIPLIB instances. All settings are able to solve 13 of 18 MIPLIB 2003 instances within the time limit. Using `ISP-NST` achieves a speed up in the solution process of 9.6%, whereas `S-orbitmin` leads to an improvement of only 1.0%; the further symmetry handling techniques allow an improvement in solution time between 9.7% (`orbi-max-s`) and 18.1% (`symre-sp`).

`ISP-NST`, however, turns out to be the best setting on both MIPLIB 2010 test sets. On M2010-sym it is able to solve 70 of 154 instances in comparison to 64 instances with the default setting which leads to a speed up of 10.9%. The orbisack and symresack settings are able to solve between 61 and 65 instances with a maximum speed up of 7.3% (`symre-p`) and a maximum slow down of 10.1% (`orbi-min-sp`). In particular, all of the `symre` and `orbi` settings that only propagate the symmetry handling inequalities lead to a speed-up between 1.9 and 7.3%, whereas separating cover inequalities always slows down the solution process. Since most of the symmetries in MIPLIB 2010 instances are either actions of the symmetric group or orbitope actions, see [47], these results indicate that it can be better to propagate symmetry handling inequalities for these kinds of symmetries then using symmetry handling inequalities.

Table 3 Comparison of different symmetry handling variants for Margot and MIPLIB instances

Setting	#nodes	Time	#opt	#act	Method-time	Sym-time
`Margot1` (16)						
default	347377.9	1070.29	6	0	1.04	0.00
orbi-max-p	196545.5	673.58	7	16	0.88	0.07
orbi-max-s	194939.3	678.27	7	16	0.87	0.07
orbi-max-sp	196327.2	673.44	7	16	0.95	0.07
orbi-min-p	196418.2	672.68	7	16	0.92	0.07
orbi-min-s	195130.9	678.33	7	16	0.90	0.07
orbi-min-sp	197467.6	674.01	7	16	0.95	0.07
symre-p	19146.5	126.70	12	16	0.83	0.06
symre-s	17399.3	133.31	13	16	0.73	0.06
symre-sp	12876.3	101.04	12	16	1.00	0.06
ISP-NST	1925.7	31.77	15	16	3.35	0.01
S-orbitmin	120704.5	436.11	9	16	0.67	0.00
`M2003-sym` (18)						
default	69260.5	369.46	13	0	1.31	0.00
orbi-max-p	68400.9	329.66	13	18	1.48	0.07
orbi-max-s	64526.8	333.52	13	18	1.40	0.07
orbi-max-sp	65883.7	328.31	13	18	1.49	0.07
orbi-min-p	68698.4	329.12	13	18	1.51	0.07
orbi-min-s	64548.7	331.27	13	18	1.44	0.07
orbi-min-sp	65854.3	327.83	13	18	1.51	0.07
symre-p	68910.8	328.98	13	18	1.41	0.07
symre-s	57820.1	311.43	13	18	1.38	0.07
symre-sp	55246.1	302.55	13	18	1.50	0.07
ISP-NST	55855.8	333.91	13	16	4.08	0.02
S-orbitmin	80703.7	365.54	13	18	1.32	0.00
`M2010-sym` (154)						
default	12829.8	929.69	64	0	0.81	0.00
orbi-max-p	12440.3	907.89	65	153	3.80	2.13
orbi-max-s	12225.0	1004.75	61	153	3.47	2.14
orbi-max-sp	12152.4	979.57	62	153	4.14	2.15
orbi-min-p	12479.5	911.46	65	153	3.80	2.16
orbi-min-s	12602.4	1023.84	61	153	3.46	2.15
orbi-min-sp	12171.0	980.92	62	153	4.08	2.15
symre-p	11860.3	861.73	65	153	3.95	2.14
symre-s	10781.2	948.33	62	153	4.02	2.15
symre-sp	11512.3	964.93	64	153	4.69	2.14
ISP-NST	7166.0	828.81	70	143	25.57	0.69
S-orbitmin	12324.7	909.62	65	153	1.59	0.00

Table 3 continued

Setting	#nodes	Time	#opt	#act	Method-time	Sym-time
M2010-bench (87)						
default	18984.6	435.61	65	0	0.73	0.00
orbi-max-p	19002.3	421.97	68	40	1.42	0.61
orbi-max-s	18023.9	408.51	68	40	1.37	0.62
orbi-max-sp	18181.2	409.96	69	40	1.39	0.61
orbi-min-p	19119.2	424.11	68	40	1.45	0.61
orbi-min-s	18510.7	416.17	69	40	1.38	0.61
orbi-min-sp	18304.7	412.36	70	40	1.39	0.61
symre-p	16931.9	388.23	69	40	1.43	0.61
symre-s	16508.7	388.20	68	40	1.36	0.61
symre-sp	16535.9	384.99	70	40	1.41	0.61
ISP-NST	13626.6	374.32	69	37	4.51	0.56
S-orbitmin	19208.5	427.82	70	40	1.32	0.00

For M2010-bench, however, the results are slightly different. The settings that propagate the symmetry handling constraints lead to a speed-up between 2.6% (`orbi-max-p`) and 10.9% (`symre-p`). But now the effect of separating minimal cover inequalities is comparable to the propagation routines; the best speed up (11.6%) is achieved by `symre-sp`, and thus, is only slightly slower than `ISP-NST` (14.1%).

Furthermore, we observe that the performance of `S-orbitmin` is comparable to the effect of the `orbi-*` settings on both MIPLIB 2010 test sets. Since most symmetries of MIPLIB 2010 instances are actions of $\mathcal{S}_n$ or orbitope actions, our results indicate that these symmetries can be efficiently handled by the simple Inequalities (19) of `S-orbitmin`. But as soon as symmetries become more complex (WB, Margot1), `S-orbitmin` is slower than specialized symmetry handling constraints.

To reduce the impact of performance variability, we applied eleven random permutations on the order of the variables of each M2010-bench instance. Table 4 shows that the fluctuation margin of the average solution time is smallest for `S-orbitmin` and the `orbi-*`-settings; altogether, the range of fluctuation is for all settings very high and lies between 80 and 90% of the average time per instance. Observe, however, that the average solution time of `S-orbitmin` on the permuted instances gets worse and the speed up decreases to 1.8%. This is not surprising, since the inequalities of `S-orbitmin` depend highly on the order of the variables. Moreover, since the `orbi-*`-settings can only handle symmetries of a subgroup of the formulation group, it is reasonable that the order of the variables has a greater effect on these settings than on techniques that can handle all symmetries. For this reason, the average speed up decreases moderately. Finally, the average speed up of `ISP-NST` and the `symre`*-settings is relatively stable, since, as argued above, these settings can handle all symmetries, and thus, depend less on the order of the variables.
We can summarize the results of this section as follows: There are problems (SONET) for which using symretopes (by the IP-formulation via symresacks) provides a huge

Table 4 Comparison of different symmetry handling variants for MIPLIB 2010 benchmark instances with 11 permuted runs

Setting	#nodes	Time	#solved inst.	Max. time diff.	Std. dev. time	Median time
default	19252.4	462.1	57	0.808	0.267	0.954
orbi-max-p	18442.2	436.8	58	0.839	0.275	0.952
orbi-max-s	17878.5	427.0	57	0.833	0.277	0.952
orbi-max-sp	17604.6	424.0	57	0.832	0.274	0.948
orbi-min-p	18503.2	437.4	58	0.864	0.283	0.946
orbi-min-s	17646.1	422.8	57	0.833	0.275	0.954
orbi-min-sp	17662.5	424.1	57	0.861	0.283	0.942
symre-p	17282.2	418.1	58	0.904	0.298	0.941
symre-s	16748.4	410.3	57	0.884	0.289	0.944
symre-sp	16680.5	409.1	56	0.893	0.292	0.941
ISP-NST	14025.5	390.4	60	0.886	0.292	0.942
S-orbitmin	18622.8	440.5	58	0.828	0.274	0.952

Column max. time diff. presents the time deviation of the fastest and the slowest run as percentage of the average time; column std. dev. time gives the standard deviation of the permuted runs as percentage of the average time. Column median time gives the median time of the permuted runs as percentage of the average time

speed-up or allow to solve problems (WB) that cannot be solved by other symmetry handling techniques. Moreover, there are instances (M2010-bench) in which symretopes are competitive with isomorphism pruning and much faster than other methods based on symmetry handling inequalities (`S-orbitmin`).

5 Conclusions

In this paper, we have studied a polyhedral approach to enforce that a binary vector is lexicographically maximal in its orbit w.r.t. a permutation group, and we successfully applied it to handle symmetry in mixed binary programs. In particular, our experiments show that the corresponding symmetry handling inequalities can help to improve the dual bound in the root node, e.g., for the wagon load balancing instances. Without adding valid inequalities for symretopes, neither of the other tested symmetry handling techniques is able to deal with these problem instances. For example, isomorphism pruning fails on these instances, because its runs out of memory while computing stabilizers of solutions.

Unfortunately, symretopes do not always lead to a strong LP relaxation. On MIPLIB instances, for example, the improvement of the dual bound of the root node is small, see Table 2. Nevertheless, symretopes can be used for these instances to speed up the solution process, although isomorphism pruning is a little faster. On average, our experiments show that both symretopes and isomorphism pruning are competitive. Consequently, our investigations show that one can use polyhedral approaches as well as techniques working directly on the enumeration tree to handle symmetry in mixed binary programs. Thus, this paper extends the tool box of efficient symmetry handling methods. Depending on the application different approaches are desirable.

However, only incorporating symmetry information via symretopes does not allow to strengthen the LP relaxation for general instances. For this reason, it is reasonable to investigate the interplay between symmetry and further problem structure. Such specialized symmetry handling inequalities were derived for some specific problems, see, e.g., Ghoniem and Sherali [17] for the wagon load balancing problem. Our aim for future research, however, is to derive generic inequalities, extending applicability.

Furthermore, a complete linear description of symretopes is known for only some very specific groups: the symmetric group and the wreath product of symmetric groups. Together with the IP-formulation of full orbitopes via orbisacks, these formulations suffice for many applications to handle symmetries. But in some applications different groups arise. In particular, for the cyclic group a characterization of the vertices of the symretope or a facet description remains open, and software experiments indicate that both seem to be rather complicated. Notably, it is unknown whether this symretope admits a small extended formulation.

Acknowledgements We thank Andreas Paffenholz for helpful discussions. Moreover, we thank two anonymous referees for their valuable suggestions that helped to improve this paper. In particular, we thank one referee for an idea to improve the running time of the algorithm in Theorem 23.

Appendix

A Visualization of the algorithm described in Theorem 23

In this section, we present an example for the application of the algorithm $\mathcal{A}$ described in the proof of Theorem 23. To this end, consider the permutation $\gamma = (1, 4)(2, 5, 7, 8, 3, 6)$ and the objective $w = (2, 1, 3, -3, -2, 1, -2, 0)^\top$. We extend w to $W = (w, \mathbb{O})$ as objective for Q_γ. Figure 3a shows which entry in the second column of a vertex of Q_γ is equal to the entry in its first column.

$$\begin{pmatrix} 1 & 4 \\ 2 & 6 \\ 3 & 8 \\ 4 & 1 \\ 5 & 2 \\ 6 & 3 \\ 7 & 5 \\ 8 & 7 \end{pmatrix} \qquad \begin{pmatrix} & \\ 0 & 0 \\ & \\ & \\ 1 & 0 \\ 0 & \\ & 1 \\ & \end{pmatrix} \qquad \begin{pmatrix} \textcolor{red}{0} & \textcolor{red}{0} \\ 0 & 0 \\ \textcolor{blue}{1} & \textcolor{blue}{1} \\ \textcolor{red}{0} & \textcolor{red}{0} \\ 1 & 0 \\ 0 & \textcolor{blue}{1} \\ & 1 \\ \textcolor{blue}{1} & \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ \textcolor{red}{0} & 1 \\ 1 & \textcolor{red}{0} \end{pmatrix}$$

(a) **(b)** **(c)** **(d)**

Fig. 3 Visualization of the optimization algorithm over Q_γ. **a** Visualization of $\gamma(X^1) = X^2$, **b** Implications of $c = 5$, **c** Row fixings above $c = 5$, **d** Fixings below $c = 5$

If $c = 4$, we want to determine a vertex x of P_γ which maximizes w and which has critical row $c = 4$ if considered as the vertex $X := (x, \gamma(x))$ of Q_γ. First, $\mathcal{A}$ detects that $X_4^1 = 1$ and $X_4^2 = 0$ because 4 is the critical row of X, and it analyzes which further entries have to be 1 or 0 due to the permutation property of the second column. Since $\gamma(4) = 1$, $\mathcal{A}$ concludes that $X_1^2 = 1$ holds. Furthermore, $X_1^1 = 1$ because row 1 is above the critical row 4 and thus, constant. Iteratively, $\mathcal{A}$ would set $X_4^2 = 1$ because $\gamma(1) = 4$. But this is a contradiction because X_4^2 is 0, since row 4 is critical. $\mathcal{A}$ detects this contradiction and terminates: row 4 cannot be critical.

If $c = 5$, $\mathcal{A}$ detects that $X_5^1 = 1$ and $X_5^2 = 0$. Since $\gamma(5) = 7$, we have $X_7^2 = 1$; similarly, because $\gamma^{-1}(5) = 2$ and $\gamma^{-2}(5) = 6$, 2 is the smallest power k of γ^{-1} such that $\gamma^{-k}(5) \geq 5$. Hence, row 2 has to be a constant 0-row and $X_6^1 = 0$ because row 5 is critical. Figure 3b illustrates the implications from critical row 5.

Then, we fix the unfixed rows above the critical row. These rows are rows 1, 3, and 4. To this end, we compute the sets

$$
\begin{aligned}
F_1 &= \{(1, 1), (4, 2), (4, 1), (1, 2)\},\\
F_3 &= \{(3, 1), (3, 2), (8, 1), (6, 2)\}, \text{ and}\\
F_4 &= F_1.
\end{aligned}
$$

Fixing any $(i, j) \in F_1$ fixes all other entries in F_1 to the same value. Because $W_1^1 + W_1^2 + W_4^1 + W_4^2 = w_1 + w_4 = -1 < 0$ we can conclude that any vertex of Q_γ which maximizes W and which has critical row 5 has 0-s in rows 1 and 4. Similarly, fixing any $(i, j) \in F_3$ fixes all other entries in F_3 to the same value. Since $W_3^1 + W_3^2 + W_8^1 + W_6^2 = 3 > 0$ the entries in F_3 of each vertex of Q_γ which maximizes W and which has critical row 5 are equal to 1. This observation is visualized in Fig. 3c.

Finally, we have to set the unfixed entries below critical row 5. The only unfixed entries are $(7, 1)$ and $(\gamma(7), 2) = (8, 2)$. Because $W_7^1 + W_8^2 = w_7 = -2 < 0$, we set $X_7^1 = X_8^2 = 0$. Now, all entries of X are fixed and a maximizer of W with critical row 5 is found, see Fig. 3d.

B Proof of Claim 3.4

In this section, we provide the missing proof of Claim 3.4

Claim Every non-trivial facet F of O_m contains m vertices of O_m whose first columns are affinely independent and are not equal to $\mathbb{1}$. Moreover, F contains m vertices of O_m whose second columns are affinely independent and are not equal to 0.

Proof Kaibel and Loos [30] showed that every non-trivial facet defining inequality of O_m is characterized by a vector $\kappa \in \{0, 1, 2, 3\}^m$ with the following properties:

- there exists $i \in [m]$ such that $\kappa_j = 0$ for all $j \in \{i + 1, \ldots, m\}$,
- $\kappa_j \neq 0$ for all $j \in [i]$, and
- $\kappa_1 = \kappa_i = 3$.

Define $\alpha = \alpha(\kappa) \in \mathbb{R}^{m\times 2}$ via

$$\alpha_j = \begin{cases} 0, & \text{if } \ j > i,\\ 1, & \text{if } \ j \in \{i-1, i\},\\ \alpha_{j+1}, & \text{if } \ j < i-1 \text{ and } \kappa_{j+1} \neq 3,\\ 2\alpha_{j+1}, & \text{if } \ j < i-1 \text{ and } \kappa_{j+1} = 3. \end{cases}$$

Then the corresponding facet defining inequality $\sum_{j=1}^{m}(a_{j1}X_{j1} + a_{j2}X_{j2}) \leq \beta$ is given by

$$(a_{j1}, a_{j2}) = \begin{cases} (0,0), & \text{if } \ \kappa_j = 0,\\ (-\alpha_j, 0), & \text{if } \ \kappa_j = 1,\\ (0, \alpha_j), & \text{if } \ \kappa_j = 2,\\ (-\alpha_j, \alpha_j), & \text{if } \ \kappa_j = 3, \end{cases}$$

and $\beta := \sum_{j\,:\,\kappa_j = 2} \alpha_j$.

In the following, it suffices to show only the first part of the claim, because the second part follows by applying the map $x \mapsto 1 - x$ to all variables, exchanging the variables of the first and second column, and flipping the meaning of $\kappa_j = 1$ and $\kappa_j = 2$. Furthermore, it is sufficient to consider only those vectors κ with $\kappa_m = 3$, cf. Loos [37, Lem. 3.62].

To prove the first part of the claim, define for every $j \in [m]$ the matrix X^j as follows. If $j < m$, define

$$(X^j_{k1}, X^j_{k2}) := \begin{cases} (1,1), & \text{if } \ k < j \text{ and } \kappa_k = 2,\\ (1,0), & \text{if } \ k = j,\\ (0,1), & \text{if } \ k > j,\\ (0,0), & \text{otherwise.} \end{cases}$$

For $j = m$, we define $(X^m_{k1}, X^m_{k2}) := (1,1)$ if $k = m$ or $\kappa_k = 2$. Otherwise, we set $(X^m_{k1}, X^m_{k2}) := (0,0)$.

These matrices are vertices of the orbisack. Moreover, the first columns of these matrices are affinely independent, because the support of the first column of X^j is contained in $[j]$ and $X^j_{j1} = 1$. By a careful consideration of all possible cases, one can show that X^j, $j \in [m]$, is contained in the facet associated with κ. In "Appendix C", we provide some examples that give an intuition why the latter statement holds. But we refrain from providing the arguments in the corresponding case distinction. □

C Illustration of the vertices constructed in Proposition 38

Table 5 illustrates the vertices constructed in the proof of Proposition 38 of the orbisack facets associated with $\kappa^1 = (3, 3, 1, 1, 3, 1, 3)$, $\kappa^2 = (3, 3, 2, 2, 3, 2, 3)$, and $\kappa^3 =$

Table 5 Vertices of some orbisack facets that are constructed in the proof of Proposition 38

Facet inequality	X^1	X^2	X^3	X^4	X^5	X^6	X^7
$\begin{pmatrix}-4&4\\-2&2\\-2&0\\-2&0\\-1&1\\-1&0\\-1&1\end{pmatrix} \leq 0$	$\begin{pmatrix}1&0\\0&1\\0&1\\0&1\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\1&0\\0&1\\0&1\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&0\\0&1\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\0&0\\1&0\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\0&0\\0&0\\1&0\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\0&0\\0&0\\0&0\\1&0\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\0&0\\0&0\\0&0\\0&0\\1&1\end{pmatrix}$
$\begin{pmatrix}-4&4\\-2&2\\0&2\\0&2\\-1&1\\0&1\\-1&1\end{pmatrix} \leq 5$	$\begin{pmatrix}1&0\\0&1\\0&1\\0&1\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\1&0\\0&1\\0&1\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&0\\0&1\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&1\\1&0\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&1\\1&1\\1&0\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&1\\1&1\\0&0\\1&0\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&1\\1&1\\0&0\\1&1\\1&1\end{pmatrix}$
$\begin{pmatrix}-4&4\\-2&2\\0&2\\-2&0\\-1&1\\0&1\\-1&1\end{pmatrix} \leq 3$	$\begin{pmatrix}1&0\\0&1\\0&1\\0&1\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\1&0\\0&1\\0&1\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&0\\0&1\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&1\\1&0\\0&1\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&1\\0&0\\1&0\\0&1\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&1\\0&0\\0&0\\1&0\\0&1\end{pmatrix}$	$\begin{pmatrix}0&0\\0&0\\1&1\\0&0\\0&0\\1&1\\1&1\end{pmatrix}$

(3, 3, 2, 1, 3, 2, 3). The first column of the table provides the facet inequality of the corresponding vector κ. The remaining columns contain the vertices $X^1, \ldots, X^7$.

D Visualization of symmetry handling decisions

In this section, we explain the steps in our procedure that adds symmetry handling inequalities to a binary program. We denote with $P(A, b, w)$ the binary program $\max\{w^\top x \,:\, Ax \leq b,\ x \in \{0,1\}^{\bar{n}}\}$, where $A \in \mathbb{R}^{m \times \bar{n}}$ and $b \in \mathbb{R}^m$ for some positive integers m and $\bar{n}$. Furthermore, we abbreviate the formulation symmetry group of $P(A, b, w)$ with $\bar{\Gamma}$.

Due to Proposition 5, we treat each factor Γ of $\bar{\Gamma}$ separately, and we classify a factor Γ as essential if Γ is acting as an orbitope on $P(A, b, w)$ or if there is $n \in \mathbb{N}$ such that $\Gamma = \mathcal{S}_n$ or $\Gamma = \mathcal{C}_n$. Otherwise, we classify the factor as non-essential. In the latter case, we handle the symmetries of Γ only if we expect that it is worthwhile to do so. Based on preliminary tests, we decided to handle non-essential symmetries if one of the following conditions hold. In this case, we set a parameter $\alpha \leftarrow 1$, otherwise we set $\alpha \leftarrow 0$.

1. If more than 95% of the variables of $P(A, b, w)$ are affected by Γ, we classify these problems as highly symmetric.
2. If the amount of affected variables lies between 5 and 95%, we handle non-essential symmetries if $\frac{\log_{10}|\Gamma|}{\text{number of generators}}$ is at most 0.55, because the number of generators in comparison to the number of elements in Γ is not too small.

Algorithm 2 visualizes the procedure that determines which symmetry handling inequalities are added to $P(A, b, w)$.

Algorithm 2: Adding symmetry handling inequalities to a symmetric binary program.

input : a factor $\Gamma \leq \mathcal{S}_n$ of the symmetry group of a binary program $P(A, b, w)$, a parameter $\alpha \in \{0, 1\}$

$i \leftarrow \min\{\ell \in [n] : \Gamma(\ell) \neq \ell\}$, $\beta \leftarrow 1$;
if $\Gamma = \mathcal{S}_n$ *or* $\Gamma = \mathcal{C}_n$ **then**
  add (19) for all $k \in \Gamma(i)$;
  $\beta \leftarrow 0$;
else if *the action of* Γ *is an orbitope action or (*$\alpha = 1$ *and a heuristic detects a subgroup* $\Gamma' \leq \Gamma$ *that acts as an orbitope)* **then**
  if *the orbits of* Γ *or* Γ' *are contained in set packing/partitioning constraints of* $Ax \leq b$ **then**
    add the packing/partitioning orbitope constraint handler of SCIP (see [32], [31]);
    $\beta \leftarrow 0$;
  else if *the orbitope contains at least three columns* **then**
    add orbisack constraints for the pairs of adjacent columns of the orbitope, cf. Proposition 29, and propagate the full orbitope;
    $\beta \leftarrow 0$;
  else
    add (19) for all $k \in \Gamma(i)$;
if *adding symresacks is allowed and* $\alpha \cdot \beta = 1$ **then**
  add symresack constraints for each generator of Γ;

References

1. Achterberg, T.: SCIP: solving constraint integer programs. Math. Program. Comput. **1**(1), 1–41 (2009)
2. Achterberg, T., Koch, T., Martin, A.: MIPLIB 2003. Oper. Res. Lett. **34**(4), 361–372 (2006)
3. Babai, L., Luks, E.M.: Canonical labeling of graphs. In: Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing, STOC '83, pp. 171–183, New York, NY, USA, ACM (1983)
4. Balas, E.: Facets of the knapsack polytope. Math. Program. **8**(1), 146–164 (1975)
5. Balas, E.: Projection, lifting and extended formulation in integer and combinatorial optimization. Ann. Oper. Res. **140**(1), 125–161 (2005)
6. Balas, E., Jeroslow, R.: Canonical cuts on the unit hypercube. SIAM J. Appl. Math. **23**(1), 61–69 (1972)
7. Chen, K.T., Fox, R.H., Lyndon, R.C.: Free differential calculus, IV. The quotient groups of the lower central series. Ann. Math. **68**, 81–95 (1958)
8. Conforti, M., Cornuéjols, G., Zambelli, G.: Extended formulations in combinatorial optimization. Ann. Oper. Res. **204**(1), 97–143 (2013)
9. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT press, Cambridge (2007)
10. Coxeter, H.S.M.: Discrete groups generated by reflections. Ann. Math. **35**(3), 588–621 (1934)
11. Crowder, H., Johnson, E.L., Padberg, M.: Solving large-scale zero-one linear programming problems. Oper. Res. **31**(5), 803–834 (1983)
12. Dias, G., Liberti, L.: Orbital Independence in Symmetric Mathematical Programs, pp. 467–480. Springer, Berlin (2015)
13. Dixon, J.D., Mortimer, B.: Permutation Groups. Graduate Texts in Mathematics, vol. 163. Springer, New York (1996)

14. Faenza, Y., Kaibel, V.: Extended formulations for packing and partitioning orbitopes. Math. Oper. Res. **34**(3), 686–697 (2009)
15. Fischetti, M., Liberti, L.: Orbital shrinking. In: Mahjoub, A.R., Markakis, V., Milis, I., Paschos, V.T. (eds.) Combinatorial Optimization. Lecture Notes in Computer Science, vol. 7422, pp. 48–58. Springer, Berlin (2012)
16. Friedman, E.J.: Fundamental domains for integer programs with symmetries. In: Dress, A., Xu, Y., Zhu, B. (eds.) Combinatorial Optimization and Applications. LNCS, vol. 4616, pp. 146–153. Springer, Berlin (2007)
17. Ghoniem, A., Sherali, H.D.: Defeating symmetry in combinatorial optimization via objective perturbations and hierarchical constraints. IIE Trans. **43**(8), 575–588 (2011)
18. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization, vol. 2. Springer, Berlin (1988)
19. Hammer, P., Johnson, E., Peled, U.: Facet of regular 0–1 polytopes. Math. Program. **8**(1), 179–206 (1975)
20. Harary, F.: On the group of the composition of two graphs. Duke Math. J. **26**(1), 29–34 (1959)
21. Harvey, W.: The fully social golfer problem. In: Smith, B.M., and Gent, I.P. (eds.) Proceedings of SymCom'03: The Third International Workshop on Symmetry in Constraint Satisfaction Problems, pp. 75–85 (2003)
22. Herr, K.: Core Sets and Symmetric Convex Optimization. Ph.D. thesis, Technische Universität Darmstadt (2013)
23. Herr, K., Rehn, T., Schürmann, A.: Exploiting symmetry in integer convex optimization using core points. Oper. Res. Lett. **41**(3), 298–304 (2013)
24. Herr, K., Rehn, T., Schürmann, A.: On lattice-free orbit polytopes. Discrete Comput. Geom. **53**(1), 144–172 (2015)
25. Hojny, C., Lüthen, H., Pfetsch, M.E.: On the size of integer programs with bounded coefficients or sparse constraints. Technical report, Optimization Online (2017). http://www.optimization-online.org/DB_HTML/2017/06/6056.html
26. IBM ILOG CPLEX Optimization Studio. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/
27. Januschowski, T., Pfetsch, M.E.: The maximum-colorable subgraph problem and orbitopes. Discrete Optim. **8**(3), 478–494 (2011)
28. Kaibel, V.: Extended formulations in combinatorial optimization. Optima **85**, 2–7 (2011)
29. Kaibel, V., Loos, A.: Branched polyhedral systems. In Eisenbrand, F., Shepherd, F.B. (eds.) Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization, IPCO 2010, Lausanne, Switzerland, 9–11 June 2010. *LNCS*, vol. 6080, pp. 177–190. Springer, Berlin (2010)
30. Kaibel, V., Loos, A.: Finding descriptions of polytopes via extended formulations and liftings. In: Mahjoub, A.R. (ed.) Progress in Combinatorial Optimization. Wiley, New York (2011)
31. Kaibel, V., Peinhardt, M., Pfetsch, M.E.: Orbitopal fixing. Discrete Optim. **8**(4), 595–610 (2011)
32. Kaibel, V., Pfetsch, M.E.: Packing and partitioning orbitopes. Math. Program. **114**(1), 1–36 (2008)
33. Koch, T., Achterberg, T., Andersen, E., Bastert, O., Berthold, T., Bixby, R.E., Danna, E., Gamrath, G., Gleixner, A.M., Heinz, S., Lodi, A., Mittelmann, H., Ralphs, T., Salvagnin, D., Steffy, D.E., Wolter, K.: MIPLIB 2010. Math. Program. Comput. **3**(2), 103–163 (2011)
34. Liberti, L.: Automatic generation of symmetry-breaking constraints. In: Yang, B., Du, D.-Z., Wang, C. (eds.) Combinatorial Optimization and Applications. Lecture Notes in Computer Science, vol. 5165, pp. 328–338. Springer, Berlin (2008)
35. Liberti, L.: Reformulations in mathematical programming: automatic symmetry detection and exploitation. Math. Program. **131**(1–2), 273–304 (2012)
36. Liberti, L., Ostrowski, J.: Stabilizer-based symmetry breaking constraints for mathematical programs. J. Glob. Optim. **60**, 183–194 (2014)
37. Loos, A.: Describing Orbitopes by Linear Inequalities and Projection Based Tools. Ph.D. thesis, University of Magdeburg (2010)
38. Maher, S.J., Fischer, T., Gally, T., Gamrath, G., Gleixner, A., Gottwald, R.L., Hendel, G., Koch, T., Lübbecke, M.E., Miltenberger, M., Müller, B., Pfetsch, M.E., Puchert, C., Rehfeldt, D., Schenker, S., Schwarz, R., Serrano, F., Shinano, Y., Weninger, D., Witt, J.T., Witzig, J.: The SCIP optimization suite 4.0. Technical report, Optimization Online (2017). http://www.optimization-online.org/DB_HTML/2017/03/5895.html
39. Margot, F.: Pruning by isomorphism in branch-and-cut. Math. Program. **94**(1), 71–90 (2002)

40. Margot, F.: Exploiting orbits in symmetric ILP. Math. Program. **98**(1–3), 3–21 (2003)
41. Margot, F.: Small covering designs by branch-and-cut. Math. Program. **94**(2), 207–220 (2003)
42. Margot, F.: Symmetric ILP: coloring and small integers. Discrete Optim. **4**(1), 40–62 (2007)
43. Margot, F.: Symmetry in integer linear programming. In: Jünger, M., Liebling, T.M., Naddef, D., Nemhauser, G.L., Pulleyblank, W.R., Reinelt, G., Rinaldi, G., Wolsey, L.A. (eds.) 50 Years of Integer Programming, pp. 647–686. Springer, Berlin (2010)
44. Ostrowski, J., Anjos, M.F., Vannelli, A.: Symmetry in Scheduling Problems. Cahier du GERAD G-2010-69, GERAD, Montreal, QC, Canada (2010)
45. Ostrowski, J., Anjos, M.F., Vannelli, A.: Modified orbital branching for structured symmetry with an application to unit commitment. Math. Program. **150**(1), 99–129 (2015)
46. Ostrowski, J., Linderoth, J., Rossi, F., Smriglio, S.: Orbital branching. Math. Program. **126**(1), 147–178 (2011)
47. Pfetsch, M.E., Rehn, T.: A computational comparison of symmetry handling methods for mixed integer programs. Technical report, Optimization Online (2015). http://www.optimization-online.org/DB_FILE/2015/11/5209.pdf
48. Rehn, T.: Exploring Core Points for Fun and Profit—a Study of Lattice-Free Orbit Polytopes. Ph.D. thesis, Universität Rostock (2013)
49. Sawada, J., Williams, A.: A gray code for fixed-density necklaces and lyndon words in constant amortized time. Theor. Comput. Sci. **502**, 46–54 (2013)
50. Schrijver, A.: Theory of Linear and Integer Programming. Wiley Interscience Series in Discrete Mathematics and Optimization. Wiley, New York (1986)
51. Sherali, H.D., Smith, J.C.: Improving discrete model representations via symmetry considerations. Manag. Sci. **47**(10), 1396–1407 (2001)
52. Tarjan, R.E.: A class of algorithms which require nonlinear time to maintain disjoint sets. J. Comput. Syst. Sci. **18**(2), 110–127 (1979)
53. Tarjan, R.E., van Leeuwen, J.: Worst-case analysis of set union algorithms. J. ACM **31**(2), 245–281 (1984)
54. van Lint, J.H., Wilson, R.M.: A Course in Combinatorics, 2nd edn. Cambridge University Press, Cambridge (1993)
55. Vande Vate, J.H.: The path set polytope of an acyclic, directed graph with an application to machine sequencing. Networks **19**(5), 607–614 (1989)