Schreier-Sims Cuts meet Stable Set: Preserving Problem Structure when Handling Symmetries

Christopher Hojny[1], Marc E. Pfetsch[2], and José Verschae[3]

[1] *Eindhoven University of Technology, Netherlands, email: c.hojny@tue.nl*
[2] *TU Darmstadt, Germany, email: pfetsch@opt.tu-darmstadt.de*
[3] *Pontificia Universidad Católica de Chile, email: jverschae@uc.cl*

November 15, 2021

Abstract

Symmetry handling inequalities (SHIs) are a popular tool to handle symmetries in integer programming. Despite their successful application in practice, only little is known about the interaction of SHIs with optimization problems. In this article, we focus on SST cuts, an attractive class of SHIs, and investigate their computational and polyhedral consequences for optimization problems. After showing that they do not increase the computational complexity of solving optimization problems, we focus on the stable set problem for which we derive presolving techniques based on SST cuts. Moreover, we derive strengthened versions of SST cuts and identify cases in which adding these inequalities to the stable set polytope maintains integrality. Preliminary computational experiments show that our techniques have a high potential to reduce both the size of stable set problems and the time to solve them.

Keywords: symmetry handling, stable set, totally unimodular

1 Introduction

The handling of symmetries in binary programs has the goal to speed up the solution process by avoiding the regeneration of symmetric solutions. To fix notation, consider the binary program $\max\{c^\top x : Ax \leq b,\ x \in \{0,1\}^n\}$, where $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$, and $c \in \mathbb{Z}^n$. Let $\mathcal{S}_n$ be the permutation group on $[n] \coloneqq \{1,\dots,n\}$. A permutation γ acts on $x \in \mathbb{R}^n$ by permuting its coordinates, i.e., $\gamma(x) \coloneqq (x_{\gamma^{-1}(1)},\dots,x_{\gamma^{-1}(n)})$. A subgroup $\Gamma \leq \mathcal{S}_n$ is a *symmetry group* of the program if every $\gamma \in \Gamma$ maps feasible solutions onto feasible solutions preserving their objective values. That is, for $x \in \mathbb{Z}^n$, $A\gamma(x) \leq b$ if and only if $Ax \leq b$, and $c^\top x = c^\top \gamma(x)$.

Different techniques have been suggested for symmetry handling such as isomorphism pruning [16, 17, 18] or adding symmetry handling inequalities (SHIs) [3, 8, 9, 10, 11, 12]; also see Margot [15] for an overview. SHIs are systems of inequalities that turn symmetric solutions infeasible, while keeping at least one (optimal) solution intact.

One particular way of handling symmetries is by the addition of inequalities based on *Schreier-Sims Tables* (SST). This has been proposed by Liberti and Ostrowski [13] and Salvagnin [22]. The main idea is that by iteratively computing group stabilizers, one can handle symmetries by adding so-called *SST cuts* of the form $x_i \leq x_j$, where variable x_i appears in the orbit of variable x_j, see Section 2 for a detailed explanation.

This approach motivates our main question:

> *What is the impact of adding SST cuts on the complexity of the underlying binary program?*

Clearly, one would hope that neither the computational nor polyhedral complexity increases. The answer to this question is not immediate in general, since SST cuts might change the structure of the underlying problem, in particular, if the problem is polynomially solvable.

In this direction, we first prove in Section 3 that computing an optimal solution that satisfies SST cuts can be done in polynomial time, if the underlying problem is solvable in polynomial time. In the remaining part of the paper, we use stable set problems (and polynomial time solvable special cases) for investigating the above question. In Section 4, we elaborate on the fact that if i and j are in a common clique, then the SST cut $x_i \leq x_j$ can be used to fix $x_i = 0$. Otherwise, the SST cut can sometimes be strengthened using cliques in the orbit of j. Our main technical contribution is to prove that if the underlying graph is trivially perfect, i.e., a laminar interval graph, then adding a carefully selected set of (strengthened) SST cuts and removing fixed variables retains total unimodularity of the constraint matrix. Hence, these SST cuts do not increase the polyhedral complexity of the problem. Interestingly, there are families of SST cuts for which total unimodularity is not preserved. In particular, this implies that different SHIs may have significant impact on the polyhedral structure of the resulting problem. We also study the computational impact of these inequalities in Section 6. The results indicate that the techniques of Section 4 are a powerful tool to reduce graph sizes and running times for symmetric stable set problems.

We note that related results as the ones in this paper can be obtained, e.g., for matching or maximum flow problems. Furthermore, Section 4 shows that SST cuts indeed preserve the structure of stable set problems. From this we derive presolving techniques that can drastically reduce the problem size. We also find that SST cuts preserve persistency of the edge relaxation, a helpful property exploited in presolving. For general independence systems, more research is need to see how our results for stable set generalize to independence systems by considering their conflict graph.

2 Schreier-Sims Table Inequalities

SST cuts are SHIs derived from Schreier-Sims tables using the following algorithm. Define the *stabilizer* $\operatorname{stab}(\Gamma, I) \coloneqq \{\gamma \in \Gamma : \gamma(i) = i \text{ for } i \in I\}$ of sets $I \subseteq [n]$ and *orbits* $\operatorname{orb}(\Gamma, i) = \{\gamma(i) : \gamma \in \Gamma\}$ for $i \in [n]$. These sets can be computed in polynomial time if Γ is given by a set of generators [23]. The algorithm performs the following steps, starting with $\Gamma' \leftarrow \Gamma$, $L \leftarrow \emptyset$: (i) select a *leader* $\ell \in [n] \setminus L$ and compute $O_\ell \leftarrow \mathrm{orb}(\Gamma', \ell)$; (ii) update $L \leftarrow L \cup \{\ell\}$, and $\Gamma' \leftarrow \mathrm{stab}(\Gamma', L)$; (iii) repeat the previous steps until Γ' becomes trivial.

We say that each element $\ell \in L$ is a *leader* and $f \in O_\ell \setminus \{\ell\}$ is a follower of ℓ. Unless stated otherwise, we relabel the leaders such that $L = \{1, \dots, k\}$, where $j \in L$ is the jth leader selected by the algorithm. One can show [13, 22] that *SST cuts*

$$-x_\ell + x_f \leq 0, \qquad\qquad \ell \in L,\ f \in O_\ell, \tag{1}$$

define a system of SHIs. We usually refer to a single cut by a pair (ℓ, f) where $\ell \in L$ and $f \in O_\ell$. Also, we define a *round* as a set of SST cuts (ℓ, f) given by a single leader $\ell \in L$ and all its followers $f \in O_\ell$. Moreover, we denote by S the set of all pairs (ℓ, f) for every $\ell \in L$ and $f \in O_\ell$.

A set S of SST cuts defines a *symmetry handling cone* via (1) that we denote by $\mathcal{C}(S)$. This cone has attained recent attention [25], for example, it has $O(n)$ facets and defines the closure of the set of vectors that are lexicographically maximal in their orbits, providing the best polyhedral approximation of lexicographically maximal vectors. In particular, every lexicographically maximal vector in $\mathbb{R}^n$ satisfies the SST cuts based on the same order of the leaders.

3 Complexity

One drawback of symmetry handling inequalities enforcing a total lexicographic order is that their separation problem is coNP-hard, cf. [7, 14]. However, SST cuts are weaker, as explained at the end of the last section. Thus, there is hope that they do not increase the computational complexity of solving a symmetry reduced problem compared to the original problem. This is indeed true:

Theorem 3.1. *Let $\mathcal{X} \subseteq \mathbb{R}^n$ and $c \in \mathbb{R}^n$. Let $\Gamma \leq \mathcal{S}_n$ be a symmetry group of the problem $(\mathcal{P})$ $\max\{c^\top x : x \in \mathcal{X}\}$. Let $S = \{(\ell, f) : \ell \in L, f \in O_\ell\}$ denote a set of SST cuts derived from Γ. If $(\mathcal{P})$ can be solved in $\mathcal{O}(T(n))$ time, then an optimal solution of the problem $(\mathcal{P}')$ $\max\{c^\top x : x \in \mathcal{X} \cap \mathcal{C}(S)\}$ can be found in $\mathcal{O}(T(n) + poly(n))$ time.*

Proof. Let x be an optimal solution of $(\mathcal{P})$. We construct an optimal solution x' of $(\mathcal{P}')$ in polynomial time. Consider the first leader $\ell = 1$ and let $i_1 \in \operatorname{argmax}\{x_i : i \in O_1\}$ and $\gamma \in \Gamma$ be such that $\gamma(i_1) = 1$. Then, $\gamma(x)$ satisfies the SST cuts $-x_1 + x_f \leq 0$ for $f \in O_1$. By replacing Γ by the stabilizer of $\ell = 1$ and x by $\gamma(x)$, we can iterate the procedure for the remaining orbits to find a point $x' \in \mathrm{orb}(\Gamma, x)$ that satisfies all SST cuts. Since x is optimal, x' is optimal too. As pointwise stabilizers can be computed in polynomial time [23], x' can be constructed in polynomial time. ☐

Note that we assume Γ to be given by a set of generators. Computing symmetries for integer programs is NP-hard [15], however, so-called *formulation* groups can be computed relatively fast in practice, see, e.g., [21].

4 Presolving Reductions

In the remainder of this article, we focus on whether SST cuts preserve problem structure. We start by investigating how the implications of SST cuts can be used in presolving routines. To this end, we consider *stable set problems*: For an undirected graph $G = (V, E)$ with node weights $c \in \mathbb{Z}^V$, find a set $I \subseteq V$ of maximal weight such that the elements in I are pairwise non-adjacent. The corresponding *edge formulation* is

$$\alpha(G) \coloneqq \max \Big\{ \sum_{v \in V} c_v \, x_v \,:\, x_u + x_v \leq 1 \;\forall \{u, v\} \in E,\; x \in \{0,1\}^V \Big\}.$$

Note that all inequalities in this formulation have $\{0, 1\}$-coefficients. Thus, adding SST cuts changes the problem structure since SST cuts have $\{0, \pm 1\}$-coefficients. To overcome this issue, we want to derive an alternative stable set problem on a graph $G' = (V', E')$ that incorporates some implications of SST cuts.

Lemma 4.1. *Let $G = (V, E)$ be an undirected graph. Let S be a set of SST cuts for $\alpha(G)$. Define $V' = V \setminus \{v \in V \,:\, v = f \text{ and } \{\ell, f\} \in E \text{ for some } (\ell, f) \in S\}$ and $G' = (V', E[V'])$, the induced subgraph. Then, $\alpha(G) = \alpha(G')$.*

Proof. Let (ℓ, f) be a leader-follower pair. If $x_f = 1$, the SST cuts imply $x_\ell = 1$ as well. Since at most one of them is contained in a stable set if $\{\ell, f\} \in E$, x_f can be fixed to 0, which is captured by G'. □

This means that we remove followers from $G = (V, E)$ that are contained in a common edge with their leaders. We call this operation the *deletion operation.*

Note that this operation does not incorporate implications of SST cuts (ℓ, f) if ℓ and f are not adjacent. To take care of this, we modify the graph G further. The *addition operation* adds $\{v, f\}$ for every neighbor v of ℓ to E. Doing so, setting $x_f = 1$ forces $x_v = 0$ for all neighbors v of ℓ.

Proposition 4.2. *Let $G = (V, E)$ be an undirected graph with weights $c \in \mathbb{Z}^V$. Let $G' = (V', E')$ arise from G by applying deletion and addition operations for a set of SST cuts. Suppose $c_v \neq 0$ for all $v \in V$. Then, every weight maximal stable set in G' is weight maximal in G and satisfies all SST cuts.*

Proof. Since the deletion operation incorporates implications of SST cuts into G', it cannot remove all optimal solutions. The missing implications of SST cuts are that setting $x_f = 1$ for a follower f implies $x_\ell = 1$ for the corresponding leader ℓ. If $x_f = 1$, then the edges introduced by the addition operation cause $x_v = 0$ for all neighbors v of ℓ. Hence, if $c_\ell > 0$, $x_\ell = 1$ in an optimal solution if $x_f = 1$. Moreover, if $c_\ell < 0$, then x_f is not set to 1 in an optimal solution, since $c_f = c_\ell < 0$ because ℓ and f are symmetric. Finally, note that setting $x_v = 1$ for some neighbor v of ℓ causes $x_f = 0$ and $x_\ell = 0$. Thus, exactly the implications of SST cuts are incorporated by the deletion and addition operation, which keeps at least one optimal solution intact. □

The previous result has an important implication for the edge formulation: SST cuts preserve persistency. Persistency is an important property, which says that if an optimal solution of the LP relaxation of the edge formulation has an integral coordinate, there exists an optimal *integral* solution of the stable set problem with the same integral coordinate [19]. This property can be used as a presolving routine for stable set problems to remove some nodes and edges. On top of this, the deletion and addition operations can be used as another symmetry-based presolving routine, *SST presolving*. While the deletion operation decreases the problem size, which is expected to have a positive impact on solving time, the addition operation introduces new edges. Since these edges handle symmetries, one might expect that the addition operation has a positive impact on solving time, which is confirmed computationally in Section 6.

5 Strengthened SST Cuts

The edge formulation is known to provide a poor LP-bound on the true weight of a maximum stable set. To strengthen this formulation, many cutting planes such as odd cycle or odd wheel inequalities have been derived [20]. One of the most important classes of inequalities, however, are *clique inequalities* $\sum_{v \in C} x_v \leq 1$ for all cliques $C \subseteq V$ in G. These inequalities define facets of the *stable set polytope* $P(G)$ if and only if C is an inclusionwise maximal clique [6]. The aim of this section is to investigate the following strengthening of SST cuts:

Lemma 5.1. *Let $G = (V, E)$ be an undirected graph and let (ℓ, f) for $f \in O_\ell$ be SST cuts derived for $P(G)$. Then, the following* SST clique cut *is an SHI for every clique $C \subseteq O_\ell$:*

$$-x_\ell + \sum_{f \in C} x_f \leq 0. \tag{2}$$

Proof. If $x_f = 1$ for some $f \in C$, the SST cuts imply $x_\ell = 1$. Since C forms a clique, at most one follower f can have $x_f = 1$, concluding the proof. □

Note that SST clique cuts generalize SST cuts, because a single follower always defines a clique.

One can show for a single round of SST cuts S that an SST clique cut defines a facet of $P(G, S) \coloneqq \operatorname{conv}\{x \in \{0,1\}^V : x \in P(G) \cap \mathcal{C}(S)\}$, the *symmetry-reduced stable set polytope*, if the clique is maximal in O_ℓ and no $f \in C$ is adjacent with ℓ. Moreover, SST clique cuts are applicable to general independence systems by defining them based on the conflict graph $\bar{G}$.

Since SST clique cuts are based on cliques, we restrict our investigation to graphs G for which $P(G)$ is completely described by clique and non-negativity inequalities: perfect graphs [5]. In general, adding SST clique cuts does not provide a complete description of $P(G, S)$, e.g., if G is a 4-cycle and S contains all possible SST cuts. Therefore, we restrict ourselves to perfect graphs G such that $P(G)$ is described by a totally unimodular constraint matrix: interval graphs [5].

An undirected graph $G = (V, E)$ is called *interval graph* if, for all $v \in V$, there is a real interval I_v such that, for all distinct u, $v \in V$, we have $\{u, v\} \in E$ if and only if $I_u \cap I_v \neq \emptyset$. A graph is called *trivially perfect (TP)* if it is an interval graph whose interval representation can be chosen to be laminar, i.e., if the intervals intersect, one is contained in the other. Let $\mathcal{C} = \mathcal{C}(G)$ be the set of maximal cliques of the undirected graph G. Then, the *clique matrix* $M(G)$ of G is the $\mathcal{C} \times V$-dimensional clique-node incidence matrix of G. The clique matrix of interval graphs, and thus trivially perfect graphs, is totally unimodular [5]. For a given graph G, let Γ_G be its automorphism group. Similarly, Γ denotes the symmetry group of the stable set program for graph G as defined in Section 4. Note that Γ is a subgroup of Γ_G whose permutations also preserve node weights.

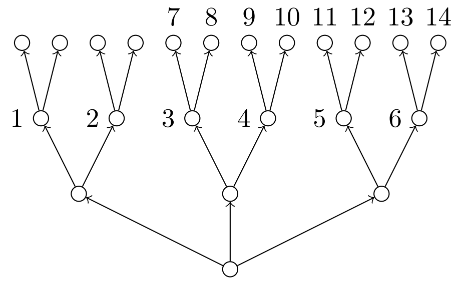

Figure 1: Example for (non-) stringent SST cuts.

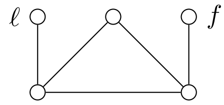

Figure 2: An interval graph and SST cut (ℓ, f).

W.l.o.g. we assume that all intervals in the interval representation $(I_v)_{v\in V}$ of a TP-graph are pairwise different. We derive a *rooted forest representation* $T_G = (V, A)$ for a TP-graph $G = (V, E)$, where $(u, v) \in A$ if and only if $I_v \subsetneq I_u$ and there is no $w \in W$ with $I_v \subsetneq I_w \subsetneq I_u$.

One natural question is whether adding SST clique cuts to a complete description of $P(G)$ provides a complete description of $P(G, S)$ if G is a TP-graph. In the remainder of this section, we give an answer by providing a sufficient criterion on when adding SST clique cuts preserves total unimodularity of the clique matrix of TP-graphs. Moreover, for TP-graphs, the number of maximal cliques is linear in the number of nodes. Picking up our motivational question from the introduction, this shows that there is a polynomial sized complete linear description of $P(G, S)$ and thus the polyhedral complexity is not increased.

To derive our sufficient criterion, we introduce the notion of stringent SST cuts. Let $L = \{1, \dots, k\}$ be the leaders of a family of SST cuts S. Note that the orbits $O_1, \dots, O_k$ define a laminar family. For $\ell \in L$, let $\mathcal{M}(\ell) \subseteq \{O_1, \dots, O_\ell\}$ be the collection of inclusionwise maximal sets in $\{O_1, \dots, O_\ell\}$. The family S of SST cuts is called *stringent* if the orbit O_ℓ of leader $\ell \in L$ is computed using the group $\operatorname{stab}(\Gamma, [\ell - 1]) \cap \operatorname{stab}(\Gamma, \mathfrak{O}_\ell)$, where $\mathfrak{O}_\ell = \bigcup_{O\in\mathcal{M}(\ell-1):\ell\notin O} O$, that is, the group must also stabilize all maximal orbits not containing ℓ.

That is, stringent SST cuts not only require to stabilize previous leaders, but also entire orbits if they do not contain the current leader.

Example 5.2. Figure 1 shows the tree representation of a TP-graph. The set of SST cuts for orbits $O_1 = \{1, \dots, 6\}$ and $O_7 = \{7, \dots, 14\}$ with leaders $L = \{1, 7\}$ (without relabeling) is not stringent, because $7 \notin O_1$. Hence, O_1 needs to be stabilized, which reduces O_7 to $\{7, 8\}$ for stringent SST cuts. Another example of stringent SST cuts is given by the leaders 1, 3, 5 (in that order) and orbits $O_1 = \{1, \dots, 6\}$, $O_3 = \{3, \dots, 6\}$, $O_5 = \{5, 6\}$, because $3, 5 \in O_1$.

Although stringency seems to be restrictive, we can implement the algorithm in Section 2 so that it always generates stringent SST cuts. Indeed, we can maintain the following property: if in Step (i) a given leader ℓ is selected, then the following leaders need to be selected from O_ℓ first. Once all elements of O_ℓ have been considered as leaders, the group $\operatorname{stab}(\Gamma, [\ell])$ stabilizes O_ℓ, and we can continue with a next leader $\ell' \in [n] \setminus O_\ell$. Hence, we obtain stringent SST cuts by choosing leaders in a depth-first search fashion.

We are now able to formulate the main result of this section.

Theorem 5.3. *Let $G = (V, E)$ be a TP-graph. Consider a set S of stringent SST cuts. The matrix that arises by applying the following two operations is totally unimodular:*

1. adding all SST clique cuts derivable from S;

2. deleting columns whose nodes get deleted by the deletion operation.

In general, this theorem does not hold if we drop stringency, because experiments with the code from [26] show that the non-stringent SST (clique) cuts from Example 1 do not preserve total unimodularity when adding them to the clique matrix of the corresponding TP-graph. Moreover, since SST clique cuts dominate SST cuts, it is necessary to replace SST cuts by clique cuts. Also, the requirement of TP-graphs and to apply the deletion operation are necessary for the validity of the theorem: Figure 2 shows an interval graph that is not TP and an SST cut such that the extended clique matrix is not totally unimodular; if there is an edge $\{\ell, f\}$ in G for an SST cut (ℓ, f), then the extended clique matrix contains a 2×2-submatrix with rows $[1, 1]$ and $[-1, 1]$, i.e., with determinant 2.

To prove Theorem 5.3, we proceed in two steps. We reduce the case of SST clique cuts to SST cuts, and then show that the result holds for this simple case.

5.1 Reduction to a Simple Case

We exploit the symmetry group structure of TP-graphs to reduce the discussion of SST clique cuts to SST cuts. Consider a TP-graph $G = (V, E)$ with automorphism group Γ_G and forest representation T. A *chain* in T is a directed path c with terminal node u such that the out-degree $\delta^+_T(v)$ with respect to T equals 1, for every node v of $c \setminus \{u\}$.

Lemma 5.4. *Let $G = (V, E)$ be a TP-graph. For any node $v \in V$, the induced subgraph of T_G in $\mathrm{orb}(\Gamma_G, v)$ decomposes into chains of the same length and Γ_G acts independently on each chain like the symmetric group.*

Proof. The nodes w in a chain c are pairwise interchangeable as exchanging their corresponding intervals I_w does not change the adjacency structure. Therefore, Γ_G acts on c as the symmetric group, while fixing the remaining nodes of G. Moreover, if a path c is not a chain, then there exist two distinct nodes u and v in c with out-degree at least 2. If v appears before u in c, the degree of v in G is larger than the degree of u. Hence, they cannot be symmetric. Therefore, for any $v \in V$, $\mathrm{orb}(\Gamma_G, v)$ decomposes into chains. They need to have the same length because the corresponding paths need to be symmetric. □

When applying SST cuts to the stable set problem, we are using a subgroup $\Gamma \leq \Gamma_G$ that also preserves node weights. In this case, we can sort the nodes along each chain consecutively by their node weights because Γ_G acts like the symmetric group on each chain. That is, Lemma 5.4 also holds for the subgroup Γ. We use this observation to define an auxiliary graph G_S for a family of SST cuts S. Whenever we compute an orbit O_ℓ, we also compute its chain decomposition according to the current stabilizer group. After computing all decompositions, G_S arises from G as follows. Each chain computed in the decomposition of the orbits is contracted into a single node. If the chain contains a leader, then we give the contracted node the lowest label of a leader within the chain. Otherwise, we give the contracted node an arbitrary label within the chain.

The interpretation of this graph is as follows. If a chain contains a leader, the deletion operation allows us to remove all nodes except for the leader from the graph. If a chain contains several leaders, it is only necessary to keep the leader considered first. For a chain c that does not contain a leader, the columns for $v \in c$ of the clique-node adjacency matrix are identical. This is true, because we never compute subchains of already considered chains, because the symmetry group acts independently like the symmetric group on each chain, i.e., we can always exchange all nodes within a chain if none of them is stabilized. They are in particular still identical if we add SST clique cuts to the matrix, because chains in T correspond to cliques in G. Hence, for deciding total unimodularity, we can remove symmetric columns.

We can now reduce Theorem 5.3 to the case of simple SST cuts by applying the following lemma.

Lemma 5.5. *Let $G = (V, E)$ be a TP-graph and let S be a set of SST clique cuts. Then, the matrix obtained by*

1. *adding SST clique cuts for S to the clique matrix $M(G)$ and*
2. *deleting columns contained in SST cuts for S such that the corresponding leader and follower are adjacent,*

is totally unimodular if and only if the matrix A_S obtained by extending $M(G_S)$ with the simple SST cuts corresponding to S in G_S is totally unimodular.

Proof. By the previous discussion, the matrix A_S is a submatrix of the extended clique matrix A of G. Thus, if A is totally unimodular, so is A_S.

For the other direction, assume A_S is totally unimodular. To see that also A is totally unimodular, select an arbitrary square submatrix B of A. If B does not contain a row corresponding to an SST clique cut, B is a submatrix of $M(G)$, and thus totally unimodular. For this reason, assume B contains a row corresponding to an SST clique cut. Select an SST clique cut in B whose leader ℓ has the largest value. Let C be the corresponding clique. If B contains two columns corresponding to nodes v and w in C, then these columns are identical by the previous discussion. Consequently, $\det(B) = 0$.

Thus, suppose B contains only one column corresponding to a node v in C. If the column corresponding to ℓ is not present in B, we expand $\det(B)$ along the row corresponding to the SST clique cut. Since this row contains exactly one 1-entry, we find $\det(B) \in \{0, \pm 1\}$ by applying the above arguments inductively. Therefore, we may assume that, for each selected SST clique cut in B, there is at most one column v that contains a node from the corresponding clique of the SST clique cut. Hence, B is a submatrix of A_S and $\det(B) \in \{0, \pm 1\}$ follows. □

5.2 Proving the Simple Case

Consequently, Theorem 5.3 holds if the following theorem holds.

Theorem 5.6. *Let $G = (V, E)$ be a TP-graph. Consider a set of stringent SST cuts for leaders $L = [k]$ and orbits $O_1, \dots, O_k$. If no orbit contains an edge from E, then the clique matrix $M(G)$ extended by the simple SST cuts is totally unimodular.*

In the remainder of this section, we prove Theorem 5.6. Let $T = T_G$ be the forest representation of a TP-graph G. We denote the set of all paths in T that connect a root node r with a leaf by $\mathcal{P}$. The paths in $\mathcal{P}$ that contain $v \in V$ are denoted by $\mathcal{P}_v$ and are in a one-to-one correspondence with the cliques $\mathcal{C}_v \subseteq \mathcal{C}$ in G that contain v. We call a set of nodes $S \subseteq V$ *path-disjoint* if $\mathcal{P}_u \cap \mathcal{P}_v = \emptyset$ for all distinct $u, v \in S$. Note that there is a one-to-one correspondence between path-disjoint sets in T and stable sets in TP-graphs G.

We define a *reduction operation* as follows: For a set $S \subseteq V$ and $v \in S$, let d be a node on the unique r-v-path, where r is the unique root of the connected component containing v in T. If we delete d from T, then T decomposes into connected components that are rooted trees. The *reduced graph* $T_d(S)$ is the graph defined by the connected components whose roots are children of d and that do not contain any node from S. We also need the following property. A family of path-disjoint sets $S_1, \dots, S_k \subseteq V$ has the *recursion property* if

(RP1) $S_1, \dots, S_k$ are pairwise disjoint, and

(RP2) for every $i \in [k-1]$, there exists $d^i \in V$ such that $S_{i+1} \subseteq T_{d^i}(\bigcup_{j=1}^{i-1} S_j)$.

If $\mathfrak{S} = \{S_1, \dots, S_k\}$ is a laminar family of subsets of V, we define, for $S \in \mathfrak{S}$, $\mathfrak{S}_S \coloneqq \{S' \in \mathfrak{S} : S' \subsetneq S\}$. We say that the laminar family $\mathfrak{S}$ has the *laminar recursion property* if

(LRP1) for all $S \in \mathfrak{S}$, there is $u_S \in S$ not contained in any set of $\mathfrak{S}_S$, and

(LRP2) the inclusionwise maximal sets in $\mathfrak{S}$ have the recursion property.

Similarly, $\mathfrak{S}$ has the (laminar) recursion property with respect to a TP-graph G, if it has the same property for its tree representation T_G. Using these concepts, we can prove Theorem 5.6. In fact, we show a stronger result for general ordering inequalities $x_u \geq x_v$ that are not necessarily based on symmetries.

Theorem 5.7. *Let $G = (V, E)$ be a TP-graph and let $S_1, \dots, S_k \subseteq V$ be stable sets satisfying the laminar recursion property. For each $i \in [k]$, let $u_i \in S_i$ adhere to (LRP1). Then, the clique matrix $M(G)$ extended by the ordering inequalities $x_{u_i} \geq x_v$ for all $i \in [k]$ and $v \in S_i \setminus \{u_i\}$ is totally unimodular.*

To prove this theorem, we need the following lemmata and concepts.

Let $\mathcal{P}$ be the set of root-leaf paths of a rooted forest $T = (V, A)$. We identify each path in $\mathcal{P}$ by its unique leaf node. For a node $v \in V$, we denote by $\operatorname{succ}(v)$ the set of direct successors of v in T, i.e., $\operatorname{succ}(v) = \{w \in V : (v, w) \in A\}$. If $P \subseteq \mathcal{P}$ is a set of paths, we denote by P_v the set of all paths containing $v \in V$. Note that, if v is a leaf, then $P_v = \{v\}$. Otherwise, $P_v = \bigcup_{w \in \operatorname{succ}(v)} P_w$.

An *equicoloring* (equitable bicoloring) of $P \subseteq \mathcal{P}$ is a partition $P^+ \cup P^-$ of P such that, for every $v \in V$, $\delta_v \coloneqq |P_v^+| - |P_v^-| \in \{0, \pm 1\}$. Due to the forest structure of T, for each $v \in V$ that is not a leaf, we have $\delta_v = \sum_{w \in \operatorname{succ}(v)} \delta_w$.

Lemma 5.8. *Let $T = (V, A)$ be a rooted tree with root r and let $P \subseteq \mathcal{P}$. Let $S \subseteq V$ be non-empty, path-disjoint and suppose that $P_v \neq \emptyset$ for each $v \in S$. Then, there exists an equicoloring $P^+ \cup P^-$ of P such that*

$$\sum_{v \in S} \delta_v \in \begin{cases} \{-1\}, & \text{if } \delta_r = -1, \\ \{0, 1\}, & \text{if } \delta_r \in \{0, 1\}. \end{cases}$$

Proof. We proceed by induction on the height h of T. If $h = 1$, then T consists just of the root node r and $S \subseteq \{r\}$. Moreover, if $r \in S$, then $S = \{r\}$ since every path in $\mathcal{P}$ contains r. In both cases, we can choose $P^- = \{r\}$ and $P^+ = \emptyset$ and the assertion holds.

If $h > 1$ and $r \notin S$, consider the forest T' that arises by removing r and all its outgoing arcs from tree T. The height of T' is $h-1$. Thus, if T' is connected, the assertion follows by induction. Otherwise, T' has $k > 1$ connected components which are rooted trees. Let $r_1, \dots, r_k$ be the corresponding root nodes and, for $i \in [k]$, let S_i be the nodes in S that are descendants of r_i. By the inductive hypothesis, we can find for every connected component $i \in [k]$ an equicoloring such that $\sum_{v \in S_i} \delta_v = \delta_{r_i}$, or $\sum_{v \in S_i} \delta_v = 0$ and $\delta_{r_i} = 1$, or $\sum_{v \in S_i} \delta_v = 1$ and $\delta_{r_i} = 0$.

Let C^- be the connected components i with $\delta_{r_i} = -1$, let C^+ be the connected components i with $\delta_{r_i} = \sum_{v \in S_i} \delta_v = 1$, let C^0 be the connected components i with $\delta_{r_i} = 1$ and $\sum_{v \in S_i} \delta_v = 0$, and let C_0 be the connected components i with $\delta_{r_i} = 0$ and $\sum_{v \in S_i} \delta_v = 1$. After possibly changing the two classes of the equicoloring for some components, we can assume $|C^+| - |C^-| \in \{0, 1\}$. Combining these equicolorings for the components in C^+ and C^- gives us an equicoloring of $C^+ \cup C^-$ with

$$\Delta \coloneqq \sum_{i \in C^+ \cup C^-} \delta_{r_i} = \sum_{i \in C^+ \cup C^-} \sum_{v \in S_i} \delta_v \in \{0, 1\}.$$

First, suppose $C_0 = \emptyset$. If $C^0 \neq \emptyset$, let $C_1, \dots, C_\ell$ be an ordering of the components in C^0 with corresponding roots $r(1), \dots, r(\ell)$. We distinguish whether $\Delta = 0$ or $\Delta = 1$. If $\Delta = 0$, we flip the two classes of the equitable partitions in C^0 with an even label; if $\Delta = 1$, we flip the classes for partitions with an odd label. Then,

$$\sum_{i=1}^{k} \delta_{r_i} = \Delta + \sum_{i=1}^{\ell} \delta_{r(i)} = \begin{cases} 0, & \text{if } \Delta = 0 \text{ and } \ell \text{ is even, or } \Delta = 1 \text{ and } \ell \text{ is odd,} \\ 1, & \text{if } \Delta = 1 \text{ and } \ell \text{ is even, or } \Delta = 0 \text{ and } \ell \text{ is odd.} \end{cases}$$

That is, $\sum_{i=1}^{k} \delta_{r_i} \in \{0, 1\}$ and $\sum_{i=1}^{k} \sum_{v \in S_i} \delta_v \in \{0, 1\}$.

Second, if $C_0 \neq \emptyset$, we proceed as before to find an equicoloring of the components in $\bar{C} \coloneqq C^+ \cup C^- \cup C^0$ with $\sum_{i \in \bar{C}} \delta_{r_i} \in \{0, 1\}$ and $\sum_{i \in \bar{C}} \sum_{v \in S_i} \delta_v \in \{0, 1\}$. Since the connected components from C_0 do not affect the value of $\sum_{i \in \bar{C}} \delta_{r_i}$, we can flip the classes of the bicoloring of every second component in C_0 to maintain the property that $\sum_{i=1}^{k} \sum_{v \in S_i} \delta_v \in \{0, 1\}$.

To conclude the proof, we extend the equicolorings of the individual connected components of T' to an equicoloring of T by associating each path in T' by the corresponding path in T. Then, for every $v \in V \setminus \{r\}$, $\delta_v \in \{0, \pm 1\}$ follows trivially. Moreover, since $\delta_r = \sum_{i=1}^{k} \delta_{r_i}$, also $\delta_r \in \{0, \pm 1\}$. In particular, δ_r has the desired relation with $\sum_{v \in S} \delta_v$ by the above argumentation. □

Lemma 5.9. *Let $T = (V, A)$ be a rooted tree, let $S_1, \dots, S_k \subseteq V$ have the recursion property, and let $P \subseteq \mathcal{P}$. If $P_v \neq \emptyset$ for each $v \in \bigcup_{i=1}^{k} S_i$, then there exists an equicoloring $P^+ \cup P^-$ of P such that $\sum_{v \in S_i} \delta_v \in \{0, \pm 1\}$ for all $i \in [k]$.*

Proof. We prove the assertion by induction on k. If $k = 1$, the statement follows from Lemma 5.8. Inductively, we can thus assume that there is an equicoloring $P^+ \cup P^-$ of P that has the desired properties for $S_1, \dots, S_{k-1}$, and show that we can adapt it to such an equicoloring for $S_1, \dots, S_k$.

Let d^{k-1} adhere to (RP2), and let $T_k = T_{d^{k-1}}(\bigcup_{i=1}^{k-2} S_i)$. W.l.o.g. we can assume that T_k consists of a single connected component. Otherwise, we show the result for the graph T' in which we replace the arcs from d^{k-1} to the roots of T_k by a single arc (d^{k-1}, d') and connect d' with the roots of T_k. The equicoloring found for T' is then also an equicoloring for T with the same properties.

Let r^k be the root of T_k. The equicoloring $P^+ \cup P^-$ derived for $S_1, \dots, S_{k-1}$ yields $\delta_{r^k} \in \{0, \pm 1\}$. If $|P_{r^k}|$ is even, then δ_{r^k} is necessarily 0 in every equicoloring. Analogously, if $|P_{r^k}|$ is odd, then $\delta_{r^k} = \pm 1$ in every equicoloring. By Lemma 5.8, there exists an equicoloring of P_{r^k} with δ^k-values such that $\sum_{v \in S_k} \delta^k_v \in \{0, \pm 1\}$. By the previous observation, $\delta^k_{r^k} = 0$ if and only if $\delta_{r^k} = 0$. Thus, after possibly flipping the two classes found in the equicoloring of P_{r^k}, $\delta^k_{r^k} = \delta_{r^k}$. Consequently, if we change the equicoloring $P^+ \cup P^-$ on P_{r^k} such that it coincides with the bicoloring found for T_k, it is still an equicoloring for T. It satisfies $\sum_{v \in S_k} \delta_v \in \{0, \pm 1\}$, and the values of $\sum_{v \in S_i} \delta_v$ for $i \in [k-1]$ did not change, because T_k does not contain any node from $\bigcup_{i=1}^{k-1} S_i$. That is, we have found the desired equicoloring. □

Proof of Theorem 5.7. We may assume that G is connected and that no node involved in an ordering inequality is the root node of the tree representation of G: Otherwise, we introduce a node w that is connected with all nodes in G, yielding a graph G'. Moreover, we can recover the assertion for G from G', because the extended clique matrix of G is a submatrix of the extended clique matrix of G'.

In the following, we work with the tree representation of G. Let $P \subseteq \mathcal{P}$ be a set of paths in the tree representation. Let D be a set of ordering inequalities encoded via the leader-follower pair (u, v), and let $U \coloneqq \{u_1, \dots, u_k\}$. That is, $D_{u,v}$ is the inequality $x_u \geq x_v$. To show that $M(G)$ extended by ordering inequalities is totally unimodular, we use Ghouila-Houri's equicoloring criterion [4]. That is, we need to find partitions $P^+ \cup P^-$ of P and $D^+ \cup D^-$ of D such that

$$\Delta_w = |P^+_w| - |P^-_w| + \sum_{u \in V} (|D^+_{u,w}| - |D^-_{u,w}|) + \sum_{v \in V} (|D^-_{w,v}| - |D^+_{w,v}|) \in \{0, \pm 1\}$$

for all $w \in V$. Our strategy is to show the statement for the case that all sets $S_1, \dots, S_k$ are pairwise disjoint first. Afterwards, we will use this result as an anchor for the general case. The anchor allows us to derive a result for the inclusionwise maximal sets among $S_1, \dots, S_k$. The corresponding equitable partition will then be modified by taking also non-maximal sets into account.

Suppose that all sets $S_1, \dots, S_k$ are pairwise disjoint. From Lemma 5.9 we derive an equicoloring $P^+ \cup P^-$ of P with $\delta_w = |P^+_w| - |P^-_w| \in \{0, \pm 1\}$ such that $\sum_{w \in S_i} \delta_w \in \{0, \pm 1\}$ for all $i \in [k]$. Note that the lemma only applies to the nodes w in S_i for which $P_w \neq \emptyset$, however, it trivially extends to the general case. In the following, we extend this equicoloring of P to an equicoloring of (P, D) by assigning a suitable partition of D. That is, we need to partition D such that

$$\Delta_w = \begin{cases} \delta_w + \sum_{v \in V} (|D^-_{w,v}| - |D^+_{w,v}|), & \text{if } w \in U, \\ \delta_w + \sum_{u \in V} (|D^+_{u,w}| - |D^-_{u,w}|), & \text{otherwise,} \end{cases}$$

is contained in $\{0, \pm 1\}$ for every $w \in V$. In particular, if w is a follower, there is a unique leader u such that its Δ-value reduces to $\delta_w = \delta_w + |D^+_{u,w}| - |D^-_{u,w}|$ by the assumption that the sets S_i are pairwise disjoint.

Note that, if $\delta_w = 1$ for some follower w, then we necessarily need to assign its leader-follower pair (u, w) to D^- to ensure $\Delta_w = 1 + |D^+_{u,w}| - |D^-_{u,w}| \in \{0, \pm 1\}$. Analogously, if $\delta_w = -1$ for some follower w, then $(u, w) \in D^+$. For followers w with $\delta_w = 0$, however, we have two choices and we will specify later on how to assign these ordering inequalities to D^+ and D^-. Denote these not yet assigned inequalities (identified by their followers $w \in V$) by $\bar{D}$.

Until now we have guaranteed that $\Delta_w \in \{0, \pm 1\}$ for all $w \in V \setminus (U \cup \bar{D})$. For a leader-follower pair (u, v), observe that assigning $(u, v) \in D$ with $\delta_v = 1$ to D^- increases Δ_u by 1, since u has a positive coefficient in the negated ordering inequality $-(-x_u + x_v \leq 0)$; analogously, assigning $(u, v) \in D$ with $\delta_v = -1$ to D^+ decreases Δ_u by 1. That is, for each $u_i \in U$, the current assignment of D^+ and D^- implies

$$\Delta_{u_i} = \delta_{u_i} + \sum_{\substack{v \in V: \\ (u_i, v) \in D}} \delta_v = \sum_{w \in S_i} \delta_w,$$

where the last equation holds since $\delta_w = 0$ for all $w \in S_i$ for which $P_w = \emptyset$. By Lemma 5.9, we thus conclude that $\Delta_{u_i} \in \{0, \pm 1\}$.

To conclude the first case, we need to assign the ordering inequalities in $\bar{D}$ to D^+ and D^-. Since $\delta_w = 0$ for each $w \in \bar{D}$, we can assign them arbitrarily to D^+ and D^- to achieve $\Delta_w = \pm 1$. The only restriction we need to take into account is the coupling of all ordering inequalities via their corresponding leaders Δ_u. Because $\Delta_u \in \{0, \pm 1\}$ if we do not consider $\bar{D}$, we can easily maintain $\Delta_u \in \{0, \pm 1\}$ by assigning the relevant ordering inequalities in $\bar{D}$ alternatingly to D^+ and D^-. Consequently, (C, D) admits an equicoloring and the assertion follows for the first case.

Note that we can choose the alternating sequence such that is has the following property, which will be exploited in the remainder of the proof: For each S_i, let $R_1, \dots, R_j \subseteq S_i \setminus \{u_i\}$ be pairwise disjoint. Then, the alternating sequence of $\bar{D}$ can be chosen such that $\sum_{w \in R_\ell} \Delta_w \in \{0, \pm 1\}$ for all $\ell \in [j]$. Indeed, this property holds, by first iterating over the elements in $R_1 \cap \bar{D}$, then over the elements in $R_2 \cap \bar{D}$, and so on.

In the second case, suppose we have relabeled the sets S_i such that $S_{\ell+1}, \dots, S_k$ are inclusionwise maximal. If we apply the previous arguments to $S_{\ell+1}, \dots, S_k$, we derive a bicoloring such that, for each $i \in \{\ell + 1, \dots, k\}$ and $w \in S_i \setminus \{u_i\}$,

$$\Delta_w = \begin{cases} 0, & \text{if } P_w \neq \emptyset, \\ \pm 1, & \text{if } P_w = \emptyset. \end{cases}$$

Moreover, if $S_1, \dots, S_j$ are the inclusionwise maximal sets among $S_1, \dots, S_\ell$, then no leader $u_{\ell+1}, \dots, u_k$ is contained in any of the sets $S_1, \dots, S_j$ by (LRP1). Thus, we can select the equicoloring such that $\sum_{w \in S_i} \Delta_w \in \{0, \pm 1\}$ for all $i \in [\ell]$ by the previously derived property.

We continue by assigning the ordering inequalities with leaders $u_1, \dots, u_j$ to D^+ and D^-. Note that this does not change the Δ-value of any node w outside $\bigcup_{i=1}^{j} S_i$. Again, if a follower w has $\Delta_w = 1$, we need to assign its ordering

Table 1: Comparison of effect of presolving of different SST variants.

	graph reductions			solving times					
orbit rule	nodes	edges	edges+	presol	cut	clique	presol+	cut+	clique+
minimum	0.90	0.81	0.85	0.55	0.56	0.56	0.43	0.50	0.47
maximum	0.80	0.60	0.65	0.36	0.29	0.30	0.25	0.31	0.30

inequality to D^- and if it has $\Delta_w = -1$ to D^+. As above, this gives the corresponding leader u_i a Δ-value of $\sum_{w \in S_i} \Delta_w$, which is 0 or ± 1 by the derived property. Hence, we can assign the remaining ordering inequalities whose follower has $\Delta_w = 0$ in an alternating order to D^+ and D^- to maintain $\Delta_{u_i} \in \{0, \pm 1\}$. We thus find an equitable partition such that, for all $i \in [j]$,

$$\Delta_w = \begin{cases} \pm 1, & \text{if } P_w \neq \emptyset, \\ 0, & \text{if } P_w = \emptyset. \end{cases}$$

Using the same arguments as above, we can proceed iteratively until we also assigned the ordering inequalities of inclusionwise minimal sets in $S_1, \dots, S_k$. □

Theorem 5.6 is now a special case of Theorem 5.7 as we sketch next.

Proof of Theorem 5.6. We briefly sketch the proof's idea. If there is no edge contained in an orbit, the orbits form stable sets in G. Moreover, the stabilizer computations guarantee that the inclusionwise maximal orbits are disjoint. Stringency implies that the inclusionwise maximal orbits have the recursion property. Since the SST leaders are not contained in succeeding orbits, the set of all orbits have the laminar recursion property. The result follows then by Theorem 5.7. □

6 Preliminary Computational Results

In this section, we discuss the impact of SST presolving, cuts, and clique cuts for the edge formulation of the maximum cardinality stable set problem. Our test set consists of all graphs from the Color02 symposium [1] and all complemented graphs from the max-clique DIMACS challenge [2] for which we could find symmetries using `SageMath 9.1` [24] within one hour. This gives us a test set of 105 graphs. For all graphs, we computed at most 50 rounds of SST cuts, where we selected an orbit of either minimal or maximal size; the leader is the variable of smallest index in each orbit.

The left part of Table 1 shows the proportion of nodes and edges that remain in the graph after applying the deletion operation of SST presolving. Column "edges+" gives the proportion of edges after additionally applying the addition operation. SST cuts based on minimum orbits reduce the number of nodes and edges by roughly 10 % and 20 %, respectively. Selecting maximum orbits even reduces these quantities by 20 % and 40 %; the biggest reduction can be achieved for the instance `latin_square_10` from Color02, where the number of nodes drops by 75 % and of edges even by 94 %. Using the addition operation increases the number of edges by five percentage points again.

In a second experiment, we investigated the impact of SST presolving and cuts on running time. These experiments have been conducted using SCIP 8.0.0.1

(githash a4eeac7) with SoPlex 5.0.1.3 as LP solver; all symmetry handling methods in SCIP have been disabled to get a fair comparison. No time limit has been imposed and all experiments were run on a Linux cluster with Intel Xeon E5 3.5 GHz quad core processors and 32 GB memory. It turns out that SCIP can solve most of these selected instances, easily even without symmetry handling. Therefore, we extracted the instances that need at least one second to be solved, which leads to a reduced test set of 26 instances.

Without any symmetry handling, the geometric mean running time is 10.3 s. The right part of Table 1 shows the proportion of solving time needed by the remaining methods for graphs obtained by the deletion operation (presol), and additionally adding SST cuts (cut) or SST clique cuts (clique). The postfix "+" indicates that we additionally apply the addition operation. To generate clique cuts, we take the set of followers F of a leader and greedily compute a clique covering of F within the subgraph induced by F. Again, the maximum orbit rule performs better. Even just applying the deletion operation reduced the running time by 64 %, adding either type of cuts reduces running time by 70 %. Additionally using the addition operation performs best and leads to a running time reduction of 75 %.

References

[1] Color02 - computational symposium: Graph coloring and its generalizations. (2002), available at http://mat.gsia.cmu.edu/COLOR02

[2] 2nd DIMACS challenge "NP-hard problems: Maximum clique, graph coloring, and satisfiability". (1992), Instances available at http://archive.dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique/

[3] 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 (2007). https://doi.org/10.1007/978-3-540-73556-4_17

[4] Ghouila-Houri, A.: Caractérisation des matrices totalement unimodulaires. C. R. Acad. Sci. Paris **254**, 1192–1194 (1962)

[5] Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Annals of Discrete Mathematics 57, Elsevier (2004)

[6] Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**, 169–197 (1981)

[7] Hojny, C.: Symmetry in Binary Programs—A Polyhedral Perspective. Ph.D. thesis, Technical University Darmstadt (2018)

[8] Hojny, C.: Packing, partitioning, and covering symresacks. Discrete Applied Mathematics **283**, 689–717 (2020). https://doi.org/10.1016/j.dam.2020.03.002

[9] Hojny, C., Pfetsch, M.E.: Polytopes associated with symmetry handling. Mathematical Programming **175**, 197–240 (2019). https://doi.org/10.1007/s10107-018-1239-7

[10] Kaibel, V., Pfetsch, M.E.: Packing and partitioning orbitopes. Mathematical Programming **114**(1), 1–36 (2008). https://doi.org/10.1007/s10107-006-0081-5

[11] Liberti, L.: Automatic generation of symmetry-breaking constraints. In: Combinatorial optimization and applications, LNCS, vol. 5165, pp. 328–338. Springer (2008). https://doi.org/10.1007/978-3-540-85097-7_31

[12] Liberti, L.: Reformulations in mathematical programming: automatic symmetry detection and exploitation. Mathematical Programming **131**(1-2), 273–304 (2012). https://doi.org/10.1007/s10107-010-0351-0

[13] Liberti, L., Ostrowski, J.: Stabilizer-based symmetry breaking constraints for mathematical programs. Journal of Global Optimization **60**, 183–194 (2014)

[14] Luks, E.M., Roy, A.: The complexity of symmetry-breaking formulas. Annals of Mathematics and Artificial Intelligence **41**(1), 19–45 (2004). https://doi.org/10.1023/B:AMAI.0000018578.92398.10

[15] 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 (2010)

[16] Margot, F.: Pruning by isomorphism in branch-and-cut. Mathematical Programming **94**(1), 71–90 (2002). https://doi.org/10.1007/s10107-002-0358-2

[17] Margot, F.: Exploiting orbits in symmetric ILP. Mathematical Programming **98**(1–3), 3–21 (2003). https://doi.org/10.1007/s10107-003-0394-6

[18] Margot, F.: Small covering designs by branch-and-cut. Mathematical Programming **94**(2), 207–220 (2003). https://doi.org/10.1007/s10107-002-0316-z

[19] Nemhauser, G.L., Trotter, L.E.: Vertex packings: Structural properties and algorithms. Mathematical Programming **8**, 232–248 (1975)

[20] Padberg, M.: On the facial structure of set packing polyhedra. Mathematical Programming **5**, 199–215 (1973)

[21] Pfetsch, M.E., Rehn, T.: A computational comparison of symmetry handling methods for mixed integer programs. Mathematical Programming Computation **11**(1), 37–93 (2019). https://doi.org/10.1007/s12532-018-0140-y

[22] Salvagnin, D.: Symmetry breaking inequalities from the Schreier-Sims table. In: van Hoeve, W.J. (ed.) Integration of Constraint Programming, Artificial Intelligence, and Operations Research. pp. 521–529. Springer, Cham (2018)

[23] Seress, A.: Permutation Group Algorithms. Cambridge University Press (2003)

[24] The Sage Developers: SageMath, the Sage Mathematics Software System (Version 9.1) (2020), `https://www.sagemath.org`

[25] Verschae, J., Villagra, M., von Niederhäusern, L.: On the geometry of symmetry breaking inequalities. In: Singh, M., Williamson, D.P. (eds.) Integer Programming and Combinatorial Optimization. pp. 73–88. Springer, Cham (2021)

[26] Walter, M., Truemper, K.: Implementation of a unimodularity test. Math. Program. Ser. C **5**(1), 57–73 (2013). https://doi.org/10.1007/s12532-012-0048-x