Math. Program., Ser. A (2013) 137:65–90
DOI 10.1007/s10107-011-0487-6

FULL LENGTH PAPER

Algorithms for highly symmetric linear and integer programs

Richard Bödi · Katrin Herr · Michael Joswig

Received: 11 January 2011 / Accepted: 29 July 2011 / Published online: 25 September 2011

Abstract This paper deals with exploiting symmetry for solving linear and integer programming problems. Basic properties of linear representations of finite groups can be used to reduce symmetric linear programming to solving linear programs of lower dimension. Combining this approach with knowledge of the geometry of feasible integer solutions yields an algorithm for solving highly symmetric integer linear programs which only takes time which is linear in the number of constraints and quadratic in the dimension.

Keywords Linear programming · Integer programming · Symmetry · Permutation group

Mathematics Subject Classification (2000) 90C10 (90C05, 52B12)

Research by Herr is supported by Studienstiftung des deutschen Volkes. Research by Joswig is partially supported by DFG Research Unit 565 "Polyhedral Surfaces" and DFG Priority Program 1489 "Experimental Methods in Algebra, Geometry, and Number Theory".

R. Bödi (✉)
School of Engineering, Zürcher Hochschule für Angewandte Wissenschaften,
Rosenstr. 2, 8400 Winterthur, Switzerland
e-mail: richard.boedi@zhaw.ch

K. Herr · M. Joswig
Fachbereich Mathematik, TU Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany
e-mail: herr@mathematik.tu-darmstadt.de

M. Joswig
e-mail: joswig@mathematik.tu-darmstadt.de

1 Introduction

It is a known fact that many standard (integer) linear programming formulations of relevant problems in optimization exhibit a lot of symmetry. In this situation a standard branch-and-cut framework repeatedly enumerates symmetric solutions, and sometimes this renders such methods useless. To address these issues the last decade saw a number of approaches to devise algorithms specialized to symmetric optimization problems. We mention a few: Margot suggests to solve symmetric integer linear programs (ILPs) via a pruned branch-and-cut approach involving techniques from computational group theory [14,15]. A recent improvement in this direction is "orbital branching" devised by Ostrowski et al. [19]. Friedman [6] as well as Kaibel and Pfetsch [9] treat symmetric ILPs by shrinking the domain of feasibility by cutting off symmetric solutions. Gatermann and Parrilo apply results from representation theory and invariant theory to semidefinite programming [7], which includes linear programming as a special case. Our approach is close in spirit to that paper. See also the survey of Margot [16] for a general overview of symmetric integer linear programming.

This is how our paper is organized: first we analyze linear programs with an arbitrary finite group of linear automorphisms. Most results in this section are known. A first key observation, Theorem 1, is that symmetric linear programming can be reduced to linear programming over the fixed space of the automorphism group. Sections 3 and 4 translate these results to the context of integer linear programming. In the sequel we concentrate on groups acting as signed permutations on the standard basis of $\mathbb{R}^n$. Section 5 contains our main contribution: our *Core Point Algorithm* B can solve an integer linear program in $\mathbb{R}^n$ whose group of linear automorphisms contains the alternating group of degree n (acting as signed permutations) in $O(mn^2)$ time, where m is the number of constraints. This is in sharp contrast with the known NP-completeness of the general integer linear programming feasibility problem. While our algorithm only works for ILPs with an exceptionally high degree of symmetry we believe that this is a first step towards an entire new class of algorithms dealing with symmetry in ILPs. Suitable generalizations are the subject to ongoing research. In Sect. 6 we discuss algorithms to determine groups of automorphisms of integer linear programs. This leads to algorithmic problems which turn out to be graph-isomorphism-complete. The final Sect. 8 contains experimental results. One of the ILP classes that we investigated computationally is motivated by work of Pokutta and Stauffer on lower bounds for Gomory-Chvátal ranks [21]. Section 7 explains the construction.

2 Automorphisms of linear programs

The purpose of this section is to introduce the notation and to collect basic facts for future reference. The results of this section up to and including Corollary 1 can be found in the literature which is why we skip some of the proofs.

We consider linear programs LP(A, b, c) of the form

$$\begin{array}{ll} \max\ c^t x & \\ \text{s.t.}\quad Ax \le b, \quad x \in \mathbb{R}^n \end{array} \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n \setminus \{0\}$. Throughout we will assume that the set $P(A, b) := \{x \in \mathbb{R}^n \mid Ax \le b\}$ of *feasible points* is not empty, and hence it is a convex polyhedron, which may be bounded or unbounded. We will also assume that an optimal solution exists. This is to say, our linear program $\mathrm{LP}(A, b, c)$ is bounded even if the feasible region may be unbounded. In this case the set of optimal solutions forms a non-empty face of $P(A, b)$. Our final assumption $c \neq 0$ is not essential for the algorithms below, but it allows to simplify the exposition somewhat.

Each row of the matrix A corresponds to one linear inequality. Suppose that one of these rows is the zero vector. Then the corresponding right hand side must be non-negative, since otherwise the linear program would be infeasible, and this was explicitly excluded above. But then this linear inequality is trivially satisfied. Therefore we will further assume that the matrix A does not contain any zero rows. In this case each row defines an affine hyperplane. This way $\mathrm{LP}(A, b, c)$ gives rise to an arrangement $\mathcal{H}(A, b)$ of m labeled affine hyperplanes in $\mathbb{R}^n$.

Definition 1 An *automorphism* of the linear program $\mathrm{LP}(A, b, c)$ is a linear transformation in $\mathrm{GL}_n\mathbb{R}$ which induces a permutation of $\mathcal{H}(A, b)$, which leaves $P(A, b)$ invariant, and which does not change the objective value $c^t x$ for any feasible point $x \in P(A, b)$.

The objective function is linear, and hence it follows that an automorphism of $\mathrm{LP}(A, b, c)$ does not change the objective value on the linear span $\mathrm{lin}(P(A, b))$ of the feasible points.

In [16], Margot considers coordinate permutations in the context of symmetric integer linear programming. This is a special case of our definition of automorphisms as coordinate permutations are induced by certain linear transformations. One could also consider more general affine transformations by additionally taking translations into account. In all what comes below this would require a number of straightforward changes. We refrain from doing so for the sake of a clearer exposition. The following examples show that the three properties to be satisfied by a linear automorphism are mutually independent.

Example 1 For $m = n = 1$ let $A = 1, b = 0$, and $c = 1$. The feasible region is the non-positive ray $\mathbb{R}_{\le 0}$. Multiplication with any positive real number γ leaves the feasible region and the hyperplane arrangement (consisting of the origin) invariant. If $\gamma \neq 1$ the objective function is not preserved.

Example 2 For $m = n = 2$ let

$$A = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad b = 0, \quad c = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

Then $P(A, b)$ is the non-negative quadrant in $\mathbb{R}^2$. Now $\gamma = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ leaves the coordinate hyperplane arrangement $\mathcal{H}(A, b)$ invariant, but it changes the feasible region. For each $x \in \mathbb{R}^2$ we have $c^t x = c^t \gamma x$.

Example 3 For $m = 3$ and $n = 2$ let

$$A = \begin{pmatrix} -1 & 0 \\ 0 & -1 \\ -1 & -2 \end{pmatrix}, \quad b = 0, \quad c = -\mathbb{1}.$$

The feasible region is the non-negative quadrant in $\mathbb{R}^2$; the third inequality is redundant. The linear transformation $\gamma = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ leaves the feasible region invariant, and it satisfies $c^t x = c^t \gamma x$ for all $x \in \mathbb{R}$. However, the hyperplane arrangement $\mathcal{H}(A, b)$ is changed.

For more examples see [3]. There it is also shown that each finite (permutation) group occurs as the group of automorphisms of a linear program.

Remark 1 It is always possible to scale the rows of the extended matrix $(A|b)$ such that the leftmost non-zero coefficient is ± 1. This allows to remove duplicate inequalities from the input by sorting. The complexity of sorting the rows by pairwise comparison is of order $O(mn \log m)$. This can be neglected in the asymptotic analysis of our algorithms below since it is always dominated. This way we can always assume that the hyperplanes in $\mathcal{H}(A, b)$, that is, the inequalities, and the rows of the extended matrix $(A|b)$ are in a one-to-one correspondence. In the rational case it is more natural to scale the inequalities to integer coefficients which are coprime. This is what we will usually do. For a more sophisticated algorithm to sort out equivalent constraints, see Bixby and Wagner [2].

Since we view points in $\mathbb{R}^n$ as column vectors, a matrix γ representing a linear transformation acts by multiplication on the left. The adjoint action on the row space, and thus on the set of linear inequalities, is by multiplication of the inverse transpose γ^{-t} on the right. The set of linear transformations permuting the arrangement $\mathcal{H}(A, b)$ forms a closed subgroup of $\mathrm{GL}_n\mathbb{R}$, that is, a linear Lie group. Similarly, the set of linear transformations leaving the feasible region $P(A, b)$ invariant forms a linear Lie group. It follows that the set $\mathrm{Aut}(\mathrm{LP}(a, b, c))$ of automorphisms of the linear program $\mathrm{LP}(A, b, c)$ also forms a linear Lie group. For basic facts about (linear) Lie groups, see Rossmann [25].

Remark 2 Clearly, the value and the set of optimal solutions of a linear program only depend on the non-redundant constraints. At the expense of one linear program per constraint one can get rid of the redundant ones. This obviously does not help to reduce the complexity of solving the linear program given since the linear program for a redundancy check is of the same size. However, for more costly algorithmic problems, like integer programming as is discussed below, this reduction can be useful. In particular, this will be the case when the group of automorphisms becomes larger, see Example 3. Notice that the notion of "invariance" from [7, Definition 3.1], specialized to linear programming, implies that redundant constraints are going to be ignored.

The following result is a consequence of the combinatorial properties of a convex polytope P: the faces of P are partially ordered by inclusion, and this partially ordered set forms a lattice. The automorphisms of this lattice, the *face lattice* of P, are called *combinatorial automorphisms*. Each linear (or affine or projective) automorphism of P induces a combinatorial automorphism, but, in general, a polytope may have many combinatorial automorphisms which are not linearly induced. See Ziegler [27] for the details.

Lemma 1 *If the feasible region $P(A, b)$ is bounded and full-dimensional, then the automorphism group $\mathrm{Aut}(\mathrm{LP}(A, b, c))$ is finite. Moreover, the objective function c satisfies $c^t \gamma x = c^t x$ for all $x \in \mathbb{R}^n$ and $\gamma \in \mathrm{Aut}(\mathrm{LP}(A, b, c))$.*

Proof Let v be a vertex of the polytope $P = P(A, b)$. Since $\dim P = n$ there are vertices $w_1, w_2, \ldots, w_n$ each of which shares an edge with v and such that the difference vectors $w_1 - v, w_2 - v, \ldots, w_n - v$ form a basis of $\mathbb{R}^n$. This implies that each combinatorial automorphism of P is induced by at most one linear automorphism. Hence the group $\mathrm{Aut}(\mathrm{LP}(A, b, c))$ is contained in the group of combinatorial automorphisms of P, which is finite. While Definition 1 asks that each *feasible* point is mapped to a (feasible) point with the same objective value, the additional claim deals with all points, feasible or not. However, this follows from $\mathrm{lin}(P(A, b)) = \mathbb{R}^n$ as c is linear. □

If the polyhedron $P(A, b)$ is not full-dimensional, then the automorphism group is a direct product of the group of automorphisms fixing the linear span of $P(A, b)$ with a full general linear group of the orthogonal complement. In the sequel we will therefore restrict our attention to the full-dimensional case.

Definition 2 Given a subset $Y \subseteq \mathbb{R}^n$ and a group $\Gamma \leq \mathrm{GL}_{\mathbb{R}}$ acting on Y, the *set of fixed points* of Y with respect to an element $\gamma \in \Gamma$ is defined by

$$\mathrm{Fix}_\gamma(Y) := \{y \in Y \mid \gamma y = y\}.$$

Therefore, the *set of fixed points* of Y with respect to Γ is given by

$$\mathrm{Fix}_\Gamma(Y) := \{y \in Y \mid \gamma y = y \quad \text{for all } \gamma \in \Gamma\} = \bigcap_{\gamma \in \Gamma} \mathrm{Fix}_\gamma(Y).$$

The set of fixed points $\mathrm{Fix}_\gamma(\mathbb{R}^n)$ is the (possibly zero-dimensional) eigenspace $\mathrm{Eig}(\gamma, 1)$ of the linear transformation γ with respect to the eigenvalue 1. This implies that $\mathrm{Fix}_\Gamma(\mathbb{R}^n)$ is a linear subspace for any group Γ of linear transformations. More generally, $\mathrm{Fix}_\Gamma(Y)$ is the intersection of this subspace with the set Y.

Remark 3 If the linear group $\Gamma \leq \mathrm{GL}_n\mathbb{R}$ is generated by the set $G \subset \Gamma$, then

$$\mathrm{Fix}_\Gamma(\mathbb{R}^n) = \bigcap_{\gamma \in G} \mathrm{Fix}_\gamma(\mathbb{R}^n) = \bigcap_{\gamma \in G} \mathrm{Eig}(\gamma, 1).$$

In particular, if G is finite, that is, if the group Γ is finitely generated, this leads to an algorithm to compute (a primal or dual basis of) the fixed space by solving one linear system of equations per transformation in the generating set G.

Remark 4 Let $\Gamma \leq \mathrm{Aut}(\mathrm{LP}(A, b, c))$ be a group of automorphisms of the linear program $\mathrm{LP}(A, b, c)$. If the feasible region $P(A, b)$ is bounded and full-dimensional then the set of fixed points $\mathrm{Fix}_\Gamma(\mathbb{R}^n)$ contains the one-dimensional linear subspace spanned by the objective vector c; see Lemma 1.

For any finite set $S \subset \mathbb{R}^n$ let

$$\beta(S) := \frac{1}{|S|} \sum_{v \in S} v$$

be its *barycenter*. The two subsequent results are basic observations from representation theory, not restricted to applications in (linear) optimization. For a brief proof, for instance, see [26, Lemma 3.5].

Lemma 2 *The map*

$$\mathbb{R}^n \to \mathrm{Fix}_\Gamma(\mathbb{R}^n), \; v \mapsto \beta(\Gamma v)$$

which sends a point to the barycenter of its Γ-orbit is a linear projection onto the fixed space.

Let $S \subseteq \mathbb{R}^n$ be a finite set which is *spanning*, that is, we require $\mathrm{lin}(S) = \mathbb{R}^n$. Further let Γ be a finite subgroup of $\mathrm{GL}_n\mathbb{R}$ acting on S: each element of Γ is a bijection if restricted to S. Phrased differently, we are considering a linear representation of an abstract group Γ on the vector space $\mathbb{R}^n$ which induces a permutation representation on the set S. In this case Γ splits S into disjoint orbits $O_1, O_2, \dots, O_k$. In our applications below, S will usually be the set of vertices of some polytope which linearly spans $\mathbb{R}^n$.

Lemma 3 *For the fixed space of Γ we have*

$$\mathrm{Fix}_\Gamma(\mathbb{R}^n) = \mathrm{lin}\{\beta(O_1), \beta(O_2), \dots, \beta(O_k)\}.$$

In particular, $\dim \mathrm{Fix}_\Gamma(\mathbb{R}^n) \leq k$.

Proof Since $S = O_1 \cup O_2 \cup \cdots \cup O_k$ is spanning and since the union of the orbits gives S it follows that

$$\mathbb{R}^n = \mathrm{lin}(O_1) + \mathrm{lin}(O_2) + \cdots + \mathrm{lin}(O_k). \tag{2}$$

For $i \in \{1, 2, \dots, k\}$ the linear subspace $\mathrm{lin}(O_i)$ is Γ-invariant. If we apply the surjective linear map $v \mapsto \beta(\Gamma v)$ from Lemma 2 to the set S, we obtain a generating set for $\mathrm{Fix}_\Gamma(\mathbb{R}^n)$. Applying the same map to a single orbit O_i similarly yields a generating set for $\mathrm{Fix}_\Gamma(\mathrm{lin}(O_i))$. Now the claim follows from the equation $\Gamma O_i = O_i$. □

Notice that the sum decomposition (2) is not necessarily direct. We now apply the results obtained so far to a finite group of automorphisms of a linear program.

Proposition 1 *Let $\Gamma \leq \mathrm{Aut}(\mathrm{LP}(A, b, c))$ be finite. If $x \in \mathbb{R}^n$ is an arbitrary point, the barycenter of its Γ-orbit satisfies $c^t \beta(\Gamma x) = c^t x$. If, moreover, $x \in P(A, b)$ is feasible, then $\beta(\Gamma x)$ is feasible, too.*

Geometrically this means that the points of one orbit are in the same affine hyperplane orthogonal to c.

Proof As the objective function is constant on the orbit Γx it follows that $c^t \beta(\Gamma x) = c^t x$. If x is a feasible point, then γx is also feasible for all $\gamma \in \Gamma$. So the barycenter $\beta(\Gamma x)$ is a convex combination of feasible points. The claim follows as the feasible region is convex. □

Since we assumed that $\mathrm{LP}(A, b, c)$ has an optimal solution, the following is an immediate consequence of the preceding result.

Corollary 1 *There exists an optimal solution of* $\mathrm{LP}(A, b, c)$ *which is a fixed point with respect to the entire automorphism group* $\mathrm{Aut}(\mathrm{LP}(A, b, c))$.

Up to minor technical details Theorem 3.3 of [7] generalizes Corollary 1 to semi-definite programming.

Let $\mathrm{LP}(A, b, c)$ be a linear program with $P(A, b)$ bounded and full-dimensional, and let $\Gamma = \langle \gamma_1, \gamma_2, \ldots, \gamma_t \rangle$ be a finite subgroup of $\mathrm{Aut}(\mathrm{LP}(A, b, c))$. Following Remark 3 we can compute a matrix E such that the kernel $\{x \mid Ex = 0\}$ is the fixed space $\mathrm{Fix}_\Gamma(\mathbb{R}^n)$: for each γ_i we determine a dual basis for the eigenspace $\{x \mid (\gamma_i - \mathrm{id})x = 0\}$ by solving a square system of linear equations. The total number of operations to do so is of order $O(tn^3)$. Throughout this paper we measure algorithmic complexity in the RAM model; that is, we ignore the encoding lengths of real numbers, and all arithmetic operations are assumed to take constant time. The group Γ acts on the rows of the extended matrix $(A|b)$, and we define a new extended matrix $(A'|b')$ by summing the rows of the same Γ-orbit. We have the following general result.

Theorem 1 *The polyhedron*

$$P' = \left\{ x \in \mathbb{R}^n \;\middle|\; A'x \leq b',\ Ex = 0 \right\}$$

is the set $\mathrm{Fix}_\Gamma(P(A, b))$ *of feasible points which is fixed under the action of Γ. In particular, $P' = \{\beta(\Gamma x) \mid x \in P(A, b)\}$. Each optimal solution of the linear program*

$$\begin{aligned} &\max\ c^t x \\ &s.t. \quad \begin{pmatrix} A' \\ E \\ -E \end{pmatrix} x \leq \begin{pmatrix} b' \\ 0 \\ 0 \end{pmatrix}, \quad x \in \mathbb{R}^n \end{aligned} \tag{3}$$

is an optimal solution of $\mathrm{LP}(A, b, c)$*, and the objective values are the same.*

Proof We constructed the matrix E in order to guarantee that each fixed point in $P = P(A, b)$ satisfies the equation $Ex = 0$. Further, each inequality of the system

$A'x \leq b'$ is a positive linear combination of valid inequalities. It follows that $\mathrm{Fix}_\Gamma(P)$ is contained in P'.

To prove the reverse inclusion consider a point x which is fixed by each transformation in Γ but which is not contained in P. Then for some index i we have the strict inequality $a_{i,\cdot}x > b_i$. Without loss of generality we can assume that the first k rows $a_{1,\cdot}, a_{2,\cdot}, \ldots, a_{k,\cdot}$ of A form the Γ-orbit of the row $a_{i,\cdot}$. It follows that $b_1 = b_2 = \cdots = b_k = b_i$. Moreover, since x is a fixed point we have

$$a_{1,\cdot}x = a_{2,\cdot}x = \cdots = a_{k,\cdot}x = a_{i,\cdot}x > b_i.$$

This implies that $(\sum_{j=1}^{k} a_{j,\cdot})x > kb_i$, and hence x is not contained in P'. We conclude that P' is the set of points in P fixed by each transformation of Γ. Now Lemma 2 says that P' is the image of P under the map $x \mapsto \beta(\Gamma x)$. The claim about the linear program (3) follows from Corollary 1. $\square$

Remark 5 It has been observed by Scharlau and Schürmann[1] that the vertices of the polyhedron P' are barycenters of orbits of vertices of P. This is a consequence of the fact that P' is the image of P under the linear map $x \mapsto \beta(\Gamma x)$.

Corollary 1 and Theorem 1 yield a direct algorithm for solving a symmetric linear program: instead of solving $\mathrm{LP}(A, b, c)$ one can solve the linear program (3). The benefit is the following: the larger the group $\Gamma \leq \mathrm{Aut}(\mathrm{LP}(A, b, c))$ the smaller the dimension of the fixed space and the number of constraints.

Remark 6 Formally, the feasible points of the derived linear program live in the same space $\mathbb{R}^n$ as the original linear program. However, an algorithm based on the Simplex Method directly benefits if the solutions are contained in a proper subspace: the rows of the matrix E describing the fixed space never have to be exchanged in a Simplex tableau. Alternatively, one can project $\mathrm{Fix}_\Gamma(\mathbb{R}^n)$ onto a full-dimensional coordinate subspace, solve the projected linear program and lift back.

In the special case where the linear program admits a group of automorphisms acting on the standard basis of $\mathbb{R}^n$ (that is, the group acts by permuting the columns) it is standard optimization practice to identify variables in the same orbit, and to solve the reduced linear program. Theorem 1 generalizes this approach to arbitrary groups of automorphisms.

3 Symmetries of integer linear programs

We now turn to our main focus. Associated with $\mathrm{LP}(A, b, c)$ is the integer linear program

$$\begin{aligned} &\max\ c^t x \\ &\text{s.t.}\quad Ax \leq b,\ x \in \mathbb{Z}^n, \end{aligned} \tag{4}$$

[1] private communication.

which we denote as $\mathrm{ILP}(A, b, c)$. Throughout we make the same assumptions as above: the linear program $\mathrm{LP}(A, b, c)$ is feasible, the matrix A does not have any zero rows, and the inequalities bijectively correspond to the hyperplane arrangement $\mathcal{H}(A, b)$; see Remark 1.

Definition 3 A *symmetry* of the integer linear program $\mathrm{ILP}(A, b, c)$ is an automorphism of $\mathrm{LP}(A, b, c)$ which acts on the signed standard basis $\{\pm e_1, \pm e_2, \ldots, \pm e_n\}$ of $\mathbb{R}^n$ as a signed permutation.

The symmetries of the integer linear program (4) form a group $\mathrm{Sym}(\mathrm{ILP}(A, b, c))$ which is a subgroup of the group $\mathrm{O}_n\mathbb{Z}$, the group of all $0/1/-1$-matrices with exactly one non-zero entry per row and column. We have $\mathrm{O}_n\mathbb{Z} = \mathrm{O}_n\mathbb{R} \cap \mathrm{GL}_n\mathbb{Z}$, and $\mathrm{O}_n\mathbb{Z}$ is isomorphic to the Coxeter group of type B_n, the group of automorphisms of the regular n-dimensional cube and its polar, the regular n-dimensional cross polytope. As a consequence, the group of symmetries of an integer linear program is always finite, even if $\mathrm{Aut}(\mathrm{LP}(A, b, c))$ is infinite.

The motivation for our definition is Lie-theoretical: let Γ be any finite subgroup of $\mathrm{GL}_n\mathbb{Z}$. Then Γ is a compact subgroup of $\mathrm{GL}_n\mathbb{R}$, hence it is contained in (a conjugate copy of) the maximal compact subgroup $\mathrm{O}_n\mathbb{R}$. It follows that, up to conjugation in $\mathrm{GL}_n\mathbb{R}$, the group Γ is a subgroup of $\mathrm{O}_n\mathbb{Z}$.

As an abstract group $\mathrm{O}_n\mathbb{Z}$ is isomorphic to the wreath product

$$\mathbb{Z}_2 \wr \mathrm{Sym}(n) = (\mathbb{Z}_2)^n \rtimes \mathrm{Sym}(n),$$

where $\mathbb{Z}_2$ is the cyclic group of order two and $\mathrm{Sym}(n)$ is the symmetric group of degree n; the group $\mathrm{Sym}(n)$ acts on the direct product $(\mathbb{Z}_2)^n$ by permuting the factors. Each element of $\mathrm{O}_n\mathbb{Z}$ can be written as a product of a sign vector and a permutation. Since a permutation is a product of disjoint cycles, each signed permutation is a product of signed cycles which are disjoint. In terms of notation we write the signs between the indices within a cycle. This is to say, $(1{+}2{-}4{+}3{-})$ denotes the signed permutation matrix

$$\begin{pmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

which is to be multiplied to column vectors from the left.

Remark 7 In the optimization literature the authors often restrict their attention to symmetries permuting the standard basis vectors; for instance, see Margot [16] and the references listed there. However, our more general analysis below shows that taking signed permutations into account does not cause any extra effort. Moreover, if the polyhedron $P(A, b)$ is full-dimensional and bounded the group of automorphisms of the linear relaxation is already finite by Lemma 1. Then $\mathrm{Aut}(\mathrm{LP}(A, b, c)) \cap \mathrm{GL}_n\mathbb{Z}$ is already contained in $\mathrm{O}_n\mathbb{Z}$ (up to conjugation in $\mathrm{GL}_n\mathbb{R}$) by the Lie-theoretical argument given above. Hence, at least in this case, considering groups of signed permutations is a natural choice.

Before we will inspect groups of symmetries of integer linear programs we need to collect a few basic results on the action of the group $\mathrm{O}_n\mathbb{Z}$ on the entire space $\mathbb{R}^n$. Throughout let Γ be a subgroup of $\mathrm{O}_n\mathbb{Z}$. Then Γ acts on the signed standard basis

$$S = \{\pm e_1, \pm e_2, \ldots, \pm e_n\}.$$

In the sequel we will always consider this particular action of Γ. There are two kinds of orbits to distinguish: the *bipolar* orbits contain at least one pair $\pm e_i$, while the *unipolar* orbits do not. Since Γ is a linear group, a signed permutation $\sigma \in \Gamma$ with $\sigma e_i = \varepsilon e_j$ and $\varepsilon \in \{\pm 1\}$ maps $-e_i$ to $-\varepsilon e_j$. Hence, a bipolar orbit only consists of pairs, that is, $-O = O$. On the other hand, for each unipolar orbit O the set $-O = \{-e_i \mid e_i \in O\}$ forms another orbit, and Γ acts equivalently on O and $-O$.

Proposition 2 *For the fixed space of Γ we have*

$$\mathrm{Fix}_\Gamma(\mathbb{R}^n) = \mathrm{lin}\,\{\beta(O) \mid O \textit{ orbit of } \Gamma\} = \mathrm{lin}\,\{\beta(O) \mid O \textit{ unipolar orbit of } \Gamma\}.$$

Proof The first equality is a consequence of Lemma 3. The second equality holds as $\beta(O) = 0$ for any bipolar orbit O. □

Remark 8 The points in S are the vertices of the regular n-dimensional cross polytope. If $O \subset S$ is a unipolar Γ-orbit, then $\beta(O)$ is the barycenter of the non-trivial face of the cross polytope which is spanned by the vertices in O. In view of cone polarity the action of Γ on S is dual to the induced action on the vertices of the regular cube $[-1, 1]$. That is, the two corresponding representations of Γ on $\mathbb{R}^n$ and on its dual space form a contra-gradient pair.

We call the action of Γ on the set of signed standard basis vectors *semi-transitive* if there are precisely two opposite orbits of length n. Moreover, we call the action *sub-transitive* if there is no proper coordinate subspace which is invariant under Γ. Clearly, a semi-transitive action is necessarily sub-transitive. The converse does not hold, but we have the following characterization.

Proposition 3 *Suppose that Γ acts sub-transitively. Then either Γ acts semi-transitively with orbits O and $-O$ such that the fixed space*

$$\mathrm{Fix}_\Gamma(\mathbb{R}^n) = \mathrm{lin}(\beta(O)) = \mathrm{lin}(\beta(-O))$$

is one-dimensional, or $\mathrm{Fix}_\Gamma(\mathbb{R}^n) = 0$.

Proof If Γ has a bipolar orbit O, then O equals the entire set S of signed standard basis vectors because Γ acts sub-transitively. In this case the fixed space reduces to the origin. If, however, each orbit is unipolar, we have exactly one pair $(O, -O)$ of opposite orbits, again due to sub-transitivity. Now the claim follows from Proposition 2. □

Corollary 2 *If Γ acts semi-transitively, then Γ is conjugate to a subgroup of* $\mathrm{Sym}(n)$ *in* $\mathrm{O}_n\mathbb{Z}$.

Proof Let O and $-O$ be the two orbits of Γ, both of which have length n. Pick a transformation $\varepsilon \in \mathrm{O}_n\mathbb{Z}$ which maps O to the standard basis $\{e_1, e_2, \ldots, e_n\}$. Now for each $\gamma \in \Gamma$ the conjugate transformation $\varepsilon\gamma\varepsilon^{-1}$ leaves the sets $\{e_1, e_2, \ldots, e_n\}$ and $\{-e_1, -e_2, \ldots, -e_n\}$ invariant. We conclude that $\varepsilon\Gamma\varepsilon^{-1}$ is a subgroup of $\mathrm{Sym}(n)$. □

We now interpret the results above for integer linear programming. Consider an integer linear program $\mathrm{ILP}(A, b, c)$ such that the set $P(A, b)$ of feasible points of the linear relaxation is full-dimensional. Let $\Gamma \leq \mathrm{Sym}(\mathrm{ILP}(A, b, c))$ be a group of automorphisms. We have $\Gamma \leq \mathrm{O}_n\mathbb{Z}$. The action of Γ on the set $\{\pm e_1, \pm e_2, \ldots, \pm e_n\}$ can be decomposed into orbits. In this way the most relevant case occurs when Γ acts sub-transitively. From Lemma 1 we know that c is contained in the fixed space $\mathrm{Fix}_\Gamma(\mathbb{R}^n)$, and then Proposition 3 says $c \neq 0$ enforces the action of Γ to be semi-transitive. Finally, by Corollary 2 we can conjugate Γ into a subgroup of $\mathrm{Sym}(n)$ acting on the standard basis $\{e_1, e_2, \ldots, e_n\}$. This is the situation that we will be dealing with in our algorithms below.

4 Layers of integer points

Our goal is to describe an algorithm for the efficient solution of a highly symmetric integer linear program. Again we consider $\mathrm{ILP}(A, b, c)$ with a group Γ of automorphisms as above.

Let us assume that the objective function $c \neq 0$ is *projectively rational*. This means that we require c to be a constant real multiple of a rational vector. For such a vector c let $\mathrm{coprime}(c)$ be the unique integral vector with coprime coefficients such that $c = \rho\, \mathrm{coprime}(c)$ for some positive real ρ. If c is a multiple of a standard basis vector, the single non-zero coefficient of $\mathrm{coprime}(c)$ is defined to be ± 1. For an integer k the *kth c-layer* is the affine hyperplane

$$H_{c,k} = \ker\,(x \mapsto c^t x) + \frac{k}{\|\mathrm{coprime}(c)\|^2}\,\mathrm{coprime}(c).$$

We have $H_{c,k} = H_{\rho c,k}$ for all $\rho > 0$, and $H_{-c,k} = -H_{c,k} = H_{c,-k}$. All points in $H_{c,k}$ attain the same value k with respect to the rescaled objective function $\mathrm{coprime}(c)$. We call k the *number* of the c-layer $H_{c,k}$. The intersection of $H_{c,k}$ with the line $\mathbb{R}c$ is called its *center*.

Lemma 4 *If $c \neq 0$ is projectively rational, the integral point $x \in \mathbb{Z}^n$ is contained in the c-layer with number* $\mathrm{coprime}(c)^t x$.

Proof The number $k = \mathrm{coprime}(c)^t x$ is an integer. We abbreviate $d = \mathrm{coprime}(c)$ and compute

$$c^t\left(\frac{k}{\|d\|^2}d\right) = c^t\left(\frac{d^t x d}{\|d\|^2}\right) = \frac{d^t d}{\|d\|^2}c^t x = c^t x.$$

Input: (A, b) such that $\mathrm{Sym}(\mathrm{ILP}(A, b, \mathbb{1}))$ acts transitively on standard basis
Output: optimal solution of $\mathrm{ILP}(A, b, \mathbb{1})$ or "infeasible"

let $z = \zeta\mathbb{1}$ be a symmetric optimal solution of the LP relaxation $\mathrm{LP}(A, b, \mathbb{1})$
$k \leftarrow \lfloor n\zeta \rfloor$
repeat
  $I \leftarrow P(A, b) \cap H_{\mathbb{1},k} \cap \mathbb{Z}^n$
  if I not empty **then**
    let x be any point in I
  else
    $k \leftarrow k - 1$
until feasible (integral) x found or $k < n\lfloor \zeta \rfloor$
return x or "infeasible"

Algorithm A: Reduction to $\mathbb{1}$-layers

Hence $x - (k/\,\|d\|^2)d$ is contained in the kernel of the linear form c^t, that is, the point x lies in the affine hyperplane $H_{c,k}$. □

For the following result it is crucial that the coefficients of $\mathrm{coprime}(c)$ are coprime.

Proposition 4 *If $c \neq 0$ is projectively rational, the c-layers $H_{c,k}$ for $k \in \mathbb{Z}$ partition the set $\mathbb{Z}^n$ of all integral points.*

Proof From Lemma 4 it is clear that each integral point is contained in some c-layer. By construction it is also obvious that the c-layers are pairwise disjoint. It remains to show that $H_{c,k} \cap \mathbb{Z}^n$ is non-empty for all $k \in \mathbb{Z}$.

Let $d = \mathrm{coprime}(c)$. Since the coefficients $d_1, d_2, \ldots, d_n$ are coprime there are integral coefficients $x_1, x_2, \ldots, x_n$ such that

$$x_1 d_1 + x_2 d_2 + \cdots + x_n d_n = \gcd(d_1, d_2, \ldots, d_n) = 1.$$

However, the left side of this equation equals $c^t x$, whence the point x is contained in the first c-layer $H_{c,1}$. Now $c^t(kx) = kc^t(x) = k$ implies that the kth layer contains the integral point kx for arbitrary $k \in \mathbb{Z}$. □

Another way of putting the statement above is that $\mathrm{coprime}(c)$ is the unique generator of the unique minimal Hilbert basis of the one-dimensional pointed cone $\mathbb{R}_{\geq 0}c$.

Remark 9 An important consequence of Proposition 4 is that for any given bounds $\ell, u \in \mathbb{R}$ there are only finitely many c-layers with feasible integral points whose objective values lie between ℓ and u. This does not hold if the objective function is not projectively rational.

Theorem 2 *For given A and b such that* $\mathrm{Sym}(\mathrm{ILP}(A, b, \mathbb{1}))$ *acts transitively on the standard basis the Algorithm A solves the integer linear program* $\mathrm{ILP}(A, b, \mathbb{1})$.

Proof Recall that throughout we assumed that the set of feasible points of the linear relaxation is bounded. Hence it cannot occur that the integer linear program is unbounded.

Let $\Gamma \leq \mathrm{Sym}(\mathrm{ILP}(A, b, \mathbb{1}))$ be a transitive group of automorphisms. The fixed space is spanned by $\mathbb{1}$. If z is an optimal solution of the relaxation $\mathrm{LP}(A, b\mathbb{1})$, then, by Proposition 1, the barycenter $\beta(\Gamma z) = \zeta\mathbb{1}$ for $\zeta = 1/n(z_1 + z_2 + \cdots + z_n)$ is also an optimal solution. Now $\lfloor\zeta\rfloor\mathbb{1}$ is an integral point in the fixed space with an objective value not exceeding the optimal value of the linear programming relaxation. Each $\mathbb{1}$-layer with a feasible integral point meets the one-dimensional polyhedron $P' = \{\beta(\Gamma x) \mid x \in P(A, b)\}$. We infer that no integral optimal solution of $\mathrm{ILP}(A, b, \mathbb{1})$ can have an objective value strictly less than $n\lfloor\zeta\rfloor$.

Due to Proposition 4 the $\mathbb{1}$-layers partition $\mathbb{Z}^n$, and so the feasible points of $\mathrm{ILP}(A, b, c)$ are contained in the set

$$\bigcup_{k=n\lfloor\zeta\rfloor}^{\lfloor n\zeta\rfloor} H_{\mathbb{1},k}.$$

□

The benefit of Algorithm A is that it reduces a (symmetric) n-dimensional integer linear programming problem to n integer feasibility problems in one dimension below. Since the latter is still an NP-complete problem not much is gained, in general. The situation changes, however, if we assume higher degrees of transitivity for the action of the group of automorphisms.

Remark 10 Searching a family of parallel affine hyperplanes for integer points as in Algorithm A also plays a key role in Lenstra's algorithm for integer linear programming which requires polynomial time in fixed dimension [12].

5 Searching integer layers efficiently

The question remaining is how to test ILP-feasibility of a c-layer in an efficient way. Our key observation is that some optimal integral solution is close to the fixed space if the integer linear program is "highly symmetric". For our purposes this property becomes manifest in the degree of transitivity of its group of symmetries: a group Γ acts *μ-transitively* on a set S if for any two μ-tuples $(s_1, \ldots, s_\mu)$, $(s_1{}', \ldots, s_\mu{}')$ of distinct elements $s_i \in S$ and $s_i{}' \in S$ there exists an element $\gamma \in \Gamma$ such that $\gamma s_i = s_i{}'$. It is immediate that a μ-transitive action is also $(\mu - 1)$-transitive. A 1-transitive action is the same as a transitive action. For very regular polytopes a high degree of transitivity on the standard basis can already imply transitivity on the vertices of the polytope. The following example plays an important role in the algorithm below: the *(r, n)-hypersimplex* $\Delta(r, n)$ is the 0/1-polytope with vertices

$$e_S = \sum_{i \in S} e_i,$$

where S ranges over all r-element subsets of $[n]$.

Proposition 5 *Let $\Gamma \leq \mathrm{GL}_n\mathbb{R}$ be a linear group which acts μ-transitively on the standard basis. Then Γ acts transitively on the set of vertices of the (r, n)-hypersimplex for any*

$$r \in \{0, 1, \ldots, \mu\} \cup \{n - \mu, n - \mu + 1, \ldots, n\}.$$

Proof By assumption Γ acts transitively on the r-element subsets of $[n]$ for $r \leq \mu$. Since Γ is a linear group it thus acts transitively on the set of vertices of $\Delta(r, n)$. The corresponding claim for the remaining hypersimplices follows since $\Delta(r, n)$ is affinely isomorphic to $\Delta(n - r, n)$ via the map $x \mapsto \mathbb{1} - x$. □

Below we will apply the previous results in the case where $\mu \geq \lfloor n/2 \rfloor + 1$. Then the group acts transitively on the sets of vertices of *all* hypersimplices. As already announced we are heading for integral optimal solutions close to the fixed space.

Definition 4 Given a c-layer with center z, an integral point in the c-layer is a *core point* if it minimizes the distance to z.

Example 4 For the objective function $c = \mathbb{1}$ and an integer $k = qn + r$ with $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, n - 1\}$, the set of core points in the kth layer consists of all integer points with r coefficients equal to $q + 1$ and $n - r$ coefficients equal to q. In particular, the number of core points in this case equals $\binom{n}{r}$. These core points are precisely the vertices of the (r, n)-hypersimplex, translated by the vector $q\mathbb{1}$.

Lemma 5 *Let $x \in P(A, b)$ be an LP-feasible point in the kth c-layer, and let $\gamma \in$ $\mathrm{Sym}(\mathrm{ILP}(A, b, c))$ with $\gamma x \neq x$. Then any point in the interior of the line segment $[x, \gamma x]$ is LP-feasible and closer to the center of the kth c-layer than x.*

Proof Since γ is an orthogonal linear map it preserves distances. The center z of the kth c-layer is fixed under γ, and this implies that $(x, z, \gamma x)$ is an isosceles triangle. We infer that $\|p - z\| < \|x - z\|$ for all points p in the interior of $[x, \gamma x]$. Since γ is an automorphism of the linear relaxation $\mathrm{LP}(A, b, c)$ the point γx is feasible, too. The feasible region is convex, and hence p is feasible. □

Theorem 3 *Suppose that $\Gamma \leq \mathrm{Sym}(\mathrm{ILP}(A, b, \mathbb{1}))$ acts $(\lfloor n/2 \rfloor + 1)$-transitively on the standard basis of $\mathbb{R}^n$, and $n \geq 2$. Then either each core point in the kth $\mathbb{1}$-layer is feasible or $H_{\mathbb{1},k}$ does not contain any feasible point.*

Proof Let x be a feasible integer point in the kth $\mathbb{1}$-layer which is not a core point. We will show that there is another feasible integer point which is closer to the center, and this will prove the claim.

Due to the invariance of $\mathbb{Z}^n$ under translation by integer vectors we may assume that $k \in \{0, 1, \ldots, n - 1\}$. Since x is not a core point, in particular, it is not the center of the kth layer. Hence x is not contained in the fixed space $\mathbb{R}\mathbb{1}$, which means that not all coordinates of x are the same. We split the set $[n]$ of coordinate directions into two subsets by considering

$$\{i \mid x_i \text{ is even}\} \quad \text{and} \quad \{i \mid x_i \text{ is odd}\}.$$

Then one of the sets—denoted by I—contains at least $\lfloor (n+1)/2 \rfloor$ elements, while the other set J has at most $\lfloor n/2 \rfloor$ elements. We will employ the $\lfloor n/2 \rfloor$-transitivity of the automorphism group to control J, and the additional degree of freedom to produce two distinct feasible integer points. We distinguish two cases.

1. Suppose that x has two coordinates $x_u \neq x_v$ such that x_u is congruent to x_v modulo two. That is, the set $\{u, v\}$ is contained in either I or J. Observe that this condition is satisfied whenever x has at least three pairwise distinct coordinates. Due to the $(\lfloor n/2 \rfloor + 1)$-transitivity of Γ there is an automorphism $\gamma \in \Gamma$ which leaves J invariant and which maps u to v. Since J is invariant, its complement $I = [n] \setminus J$ is invariant, too. Notice that we do not require the set J to be non-empty (if $\{u, v\} \subseteq I$).
   Letting $x' = \gamma x$ we observe that x_i and x'_i are congruent modulo two for all $i \in [n]$. Since $x_u \neq x_v = x_{\gamma(u)} = x'_u$ we have $x \neq x'$, and hence
   $$y = \frac{1}{2}(x + x') = \frac{1}{2}(x + \gamma x)$$
   is an integer point in the interval $[x, \gamma x]$.
2. Otherwise the n coordinates of the point x attain exactly two distinct values,one of them being even, the other one odd. Let x_u and x_v be two coordinates with distinct values. Without loss of generality, $x_i = x_u$ for all $i \in I$ and $x_j = x_v$ for all $j \in J$. Due to the transitivity of Γ there is an automorphism $\gamma \in \Gamma$ with $\gamma e_u = e_v$. Then x and γx are distinct points. Consider an interior point
   $$y = \lambda x + (1 - \lambda)\gamma x \quad \text{for } 0 < \lambda < 1 \tag{5}$$
   in the line segment $[x, \gamma x]$. We want to find a parameter λ such that y is integral. As x has only two distinct coordinates the ith coordinate of y can attain the following values only:
   $$y_i = \lambda x_u + (1 - \lambda) x_u = x_u \quad \text{or} \tag{6}$$
   $$y_i = \lambda x_v + (1 - \lambda) x_v = x_v \quad \text{or} \tag{7}$$
   $$y_i = \lambda x_u + (1 - \lambda) x_v = \lambda(x_u - x_v) + x_v \quad \text{or} \tag{8}$$
   $$y_i = \lambda x_v + (1 - \lambda) x_u = \lambda(x_v - x_u) + x_u. \tag{9}$$
   Since x is integral the coordinates of types (6) and (7) are integers for arbitrary parameters $\lambda \in (0, 1)$. The coordinates of types (8) and (9) are integral if $\lambda \cdot |x_u - x_v| \in \mathbb{Z}$.
   We assumed that x is contained in the kth $\mathbb{1}$-layer for some $k = 0, 1, \ldots, n-1$ and that it is not a core point. In Example 4 it has been observed that the core points in these layers are the vertices of a translated hypersimplex. We learned

Input: (A, b) such that $\mathrm{Sym}(\mathrm{ILP}(A, b, \mathbb{1}))$ acts $(\lfloor n/2 \rfloor + 1)$-transitively on standard basis
Output: optimal solution of $\mathrm{ILP}(A, b, \mathbb{1})$ or "infeasible"
let $z = \zeta \mathbb{1}$ be a symmetric optimal solution of the LP relaxation $\mathrm{LP}(A, b, \mathbb{1})$
$d \leftarrow \lfloor n\zeta \rfloor - n\lfloor \zeta \rfloor$
repeat
  $x \leftarrow (\underbrace{\lfloor \zeta \rfloor + 1, \dots, \lfloor \zeta \rfloor + 1}_{d}, \underbrace{\lfloor \zeta \rfloor, \dots, \lfloor \zeta \rfloor}_{n-d})$
  if x infeasible **then**
    $d \leftarrow d - 1$
until feasible x found or $d < 0$
return x or "infeasible"

Algorithm B: Core point algorithm

that some coordinate difference $|x_i - x_k|$ must exceed one, as x is not a core point. Since all coefficients are equal to either x_u or x_v it follows that $|x_u - x_v| \geq 2$. We can now set

$$\lambda = \frac{1}{|x_u - x_v|}$$

in the formula (5).

In both cases we obtain an integral point y in the interior of the interval $[x, \gamma x]$. By Lemma 5, such a point is always closer to the center than x. This shows that there exists a feasible core point in the same layer as x. Applying Proposition 5 with $\mu = \lfloor n/2 \rfloor + 1$ yields that then *each* core point must be feasible. □

Now Algorithm A can be modified in Step 6 to check a single core point per layer for feasibility, provided that the group of automorphisms of the ILP acts at least $(\lfloor n/2 \rfloor + 1)$-transitively. This is our *Core Point Algorithm* B.

Corollary 3 *For given A and b such that* $\mathrm{Sym}(\mathrm{ILP}(A, b, \mathbb{1}))$ *acts* $(\lfloor n/2 \rfloor + 1)$*-transitively on the standard basis the Core Point Algorithm B solves the integer linear program* $\mathrm{ILP}(A, b, \mathbb{1})$ *in* $O(mn^2)$ *time.*

Proof The correctness follows from Theorems 2 and 3. The main loop of the algorithm is executed at most n times. In each step the costs are dominated by checking one point in $\mathbb{R}^n$ for feasibility against m linear inequalities. □

Remark 11 The linear search in Algorithms A and B cannot be substituted by a direct bisectional approach. The reason is that the set of all k in $\{0, 1, \dots, \lfloor n\zeta \rfloor - n\lfloor \zeta \rfloor\}$ such that the kth $\mathbb{1}$-layer contains a feasible point is not necessarily (the set of integer points of) an interval.

Remark 12 It is a classical result that a group acting $(\lfloor n/2 \rfloor + 1)$-transitively on n elements is necessarily isomorphic to the alternating group A_n or to the full symmetric group S_n; see Miller [18] and also Cameron [5] for a modern view on multiply transitive group actions. Notice, however, that S_5 (which is isomorphic, as an abstract group, to the projective linear group $\mathrm{PGL}_2(5)$) admits an exceptional 3-transitive action on six elements. This does not meet the requirements for Theorem 3 or its Corollary 3.

6 Finding all symmetries

For the algorithms presented it is never necessary to know the entire group of automorphisms of $\mathrm{LP}(A, b, c)$ or $\mathrm{ILP}(A, b, c)$. Generally, any subgroup will do, the larger the better. Yet here we would like to discuss the question of how to find automorphisms of integer linear programs. From the input data we will construct a labeled graph $\mathrm{G}(A, b, c)$ whose group of labeled automorphisms coincides with the group $\mathrm{Sym}(\mathrm{ILP}(A, b, c))$.

Expressing symmetry in optimization via graph automorphisms is not a new idea: the linear automorphism group of a polytope and of a linear program can be obtained by computing the automorphism group of a certain graph as described by Bremner et al. [4]. The combinatorial automorphisms of a polytope are the (labeled) graph automorphisms of the bipartite graph encoded by the vertex-edge-incidences. This directly follows from the fact that the face lattice of a polytope is atomic and coatomic; see Kaibel and Schwartz [10]. Liberti studies automorphisms of optimization problems which are more general than integer linear programs [13]. His approach, however, deals with expression trees obtained from a specific encoding of the optimization problem. None of these concepts seems to be directly related to the kind of symmetry studied here. Ideas similar to ours, however, have been applied by Puget [22] for symmetry detection in constraint programming; see also [16, §3] and, additionally, Berthold and Pfetsch [1] for symmetries of 0/1-ILPs.

The complexity status of the graph isomorphism problem is notoriously open. While the known algorithms for determining the automorphism group of a graph require exponential time, there exist software packages, for instance, `nauty` [17] or `SymPol` [24], that can solve such problems very well in practice.

For a given matrix $A \in \mathbb{R}^{m\times n}$, right hand side $b \in \mathbb{R}^m$, and objective function $c \in \mathbb{R}^n$ we will now associate two undirected simple graphs, the *ILP graph* $\mathrm{G}(A, b, c)$, and the *reduced ILP graph* $\mathrm{G}'(A, b, c)$. For the sake of a simplified exposition we start out by describing the reduced ILP graph. Throughout we assume that the rows of the extended matrix $(A|b)$ are normalized as described in Remark 1. We have one node α_{ij} for each position in the matrix A, one node ρ_i for each row, and one node ζ_j for each column, that is, $(i, j) \in [m] \times [n]$, where $[n] = \{1, 2, \dots, n\}$. Further, we have one node κ_u for each distinct coefficient u in the matrix A, one node λ_v for each distinct coefficient v of b, and one node μ_w for each distinct coefficient w of c. This gives a total of $mn + m + n + n_A + n_b + n_c$ nodes, where n_A, n_b, and n_c denotes the respective number of different entries in A, b, and c. The nodes receive labels in the following way: all positions share the same label, the rows receive a second, and the columns a third label. Each node corresponding to one of the coefficients receives an individual label. This way we arrive at $n_A + n_b + n_c + 3$ labels altogether. The edges of $\mathrm{G}'(A, b, c)$ are defined as follows: the node α_{ij} is adjacent to ρ_i and ζ_j as well as to the coefficient node which represents the coefficient a_{ij} of the matrix A. Moreover, the row node ρ_i is adjacent to the node λ_{b_i}, and the node ζ_j is adjacent to the node μ_{c_j}. This totals to $3mn + m + n$ edges.

To the best of our knowledge the above construction of (reduced) ILP graphs is new. However, it bears some similarity with a construction due to Puget [22]. There

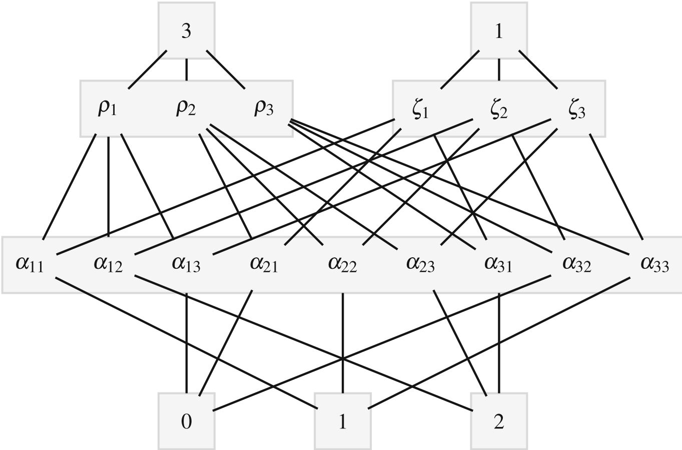

Fig. 1 The reduced ILP graph for (10)

the author considers constraint satisfiability problems with a finite choice for each variable.

Example 5 The reduced ILP graph of the integer linear program

$$\begin{array}{ll} \max & x_1 + x_2 + x_3 \\ \text{s.t.} & x_1 + 2x_2 \le 3 \\ & x_2 + 2x_3 \le 3 \\ & 2x_1 + x_3 \le 3, \quad x_i \in \mathbb{Z} \end{array} \tag{10}$$

is shown in Fig. 1.

Let γ be an automorphism of $\mathrm{G}'(A, b, c)$ which respects all node labels. Since the common label of the column nodes is preserved γ induces a column permutation $\psi_\gamma \in \mathrm{Sym}(n)$. Now ψ_γ acts on the standard basis $\{e_1, e_2, \ldots, e_n\}$, and by linear extension we obtain a linear transformation which we denote ψ_γ^*.

Lemma 6 *The linear transformation ψ_γ^* is a symmetry of* $\mathrm{ILP}(A, b, c)$.

Proof As above let γ be a labeled automorphism of $\mathrm{G}' = \mathrm{G}'(A, b, c)$ with induced column permutation $\psi = \psi_\gamma$ and linear transformation $\psi^* \in \mathrm{Sym}(n) \le \mathrm{GL}_n\mathbb{R}$. As for the column nodes the graph automorphism γ also induces a permutation $\phi \in \mathrm{Sym}(m)$ of the row nodes of G'. The position nodes α_{ij} form a label class of their own, and so they are permuted by γ as well. Since each position node is adjacent to precisely one row and one column node we infer that $\gamma(\alpha_{ij}) = \alpha_{\phi(i),\psi(j)}$. Each position node is adjacent to precisely one matrix coefficient node, each of which forms a singleton label class. This implies that the coefficient a_{ij} corresponding to the node α_{ij} is the same as the coefficient $a_{\phi(i),\psi(j)}$. Likewise we obtain $b_i = b_{\phi(i)}$ and $c_j = c_{\psi(j)}$. This means that ψ_γ^* is a symmetry of $\mathrm{ILP}(A, b, c)$. □

Proposition 6 *The map $\gamma \mapsto \psi^*_\gamma$ is an isomorphism from the group of labeled automorphisms of the graph* $\mathrm{G}'(A, b, c)$ *to the group* $\mathrm{Sym}(\mathrm{ILP}(A, b, c)) \cap \mathrm{Sym}(n)$.

Proof We describe the inverse map. To this end let σ be a symmetry of $\mathrm{ILP}(A, b, c)$ which acts on the standard basis of $\mathbb{R}^n$. Hence σ induces a permutation ϕ of the rows of the extended matrix $(A|b)$ and a permutation ψ of the columns of A. It is obvious how ϕ and ψ induce permutations of the row nodes and of the column nodes of G'. By the same reasoning as in the proof of Lemma 6 the pair (ϕ, ψ) uniquely extends to a labeled graph automorphism $\gamma(\sigma)$ of the reduced ILP graph.

We omit the straightforward proofs that the equations $\gamma(\psi^*_\gamma) = \gamma$ and $\psi^*_{\gamma(\sigma)} = \sigma$ both hold. From these it follows that the map $\gamma \mapsto \psi^*_\gamma$ is bijective. In both groups the multiplications are given by concatenations of maps. A direct computation yields $\psi^*_{\gamma_1\gamma_2} = \psi^*_{\gamma_1}\psi^*_{\gamma_2}$; all maps are acting on the left. Hence the group structures are preserved. □

We now explain how the full ILP graph $\mathrm{G}(A, b, c)$ differs from the reduced ILP graph $\mathrm{G}'(A, b, c)$. The key to the construction of $\mathrm{G}'(A, b, c)$ is the map $\gamma \mapsto \psi^*_\gamma$ yielding a linear transformation which acts as a permutation of the standard basis of $\mathbb{R}^n$. In order to allow for signed permutations certain nodes have to be duplicated: each column node ζ_j in $\mathrm{G}'(A, b, c)$ gets a *twin node* $\hat{\zeta}_j$ in $\mathrm{G}(A, b, c)$, each matrix coefficient node α_{ij} corresponding to a non-zero coefficient gets a twin node $\hat{\alpha}_{ij}$. Moreover, we add further nodes representing negatives of non-zero coefficients in the matrix A and the objective function c unless nodes with these labels already exist. This way $\mathrm{ILP}(A, b, c)$ has less than twice as many nodes as $\mathrm{ILP}'(A, b, c)$; it is always strictly less as the nodes corresponding to the coefficients in b are never duplicated. We also add edges such that first $\hat{\alpha}_{ij}$ is adjacent to ρ_i and $\hat{\zeta}_j$ for all i and j, second $\hat{\zeta}_j$ is adjacent to μ_{-c_j}, third $\hat{\alpha}_{ij}$ is adjacent to $\kappa_{-a_{ij}}$, and, finally, the twins are matched up: α_{ij} is adjacent to $\hat{\alpha}_{ij}$ and $\hat{\zeta}_j$ is adjacent to ζ_j. The labeling is extended in a way such that twins share the same label; the nodes newly introduced for negatives of coefficients receive new singleton labels.

Each labeled graph automorphism of $\mathrm{G}'(A, b, c)$ uniquely extends to a labeled graph automorphism of $\mathrm{G}(A, b, c)$, but the automorphism group of the non-reduced ILP graph is larger, in general. We have the following result.

Theorem 4 *The group of labeled graph automorphisms of* $\mathrm{G}(A, b, c)$ *is isomorphic to the group of symmetries of* $\mathrm{ILP}(A, b, c)$.

Proof One can follow the strategy in the proof of Proposition 6. We know that a labeled graph automorphism of $\mathrm{G}'(A, b, c)$ encodes a symmetry of $\mathrm{ILP}(A, b, c)$ which permutes the set $\{e_1, e_2, \ldots, e_n\}$. Now a labeled graph automorphism of $\mathrm{G}(A, b, c)$ may map a column node ζ_j to some node $\hat{\zeta}_k$. But then it follows that $\hat{\zeta}_j$ is mapped to ζ_k since $\hat{\zeta}_j$ is the only column node adjacent to ζ_j, and ζ_k is the only column node adjacent to $\hat{\zeta}_k$. This shows that the permutation of the column nodes can be extended to a linear transformation. As in the proof of Proposition 6 one can show that this linear transformation is a symmetry of the integer linear program. Conversely, each such symmetry acts like a signed permutation on the signed standard basis and yields a labeled isomorphism of the graph $\mathrm{G}(A, b, c)$. □

Roughly speaking, a class $\mathcal{C}$ of graphs is *graph isomorphism complete* if the problem of deciding whether any two given graphs in $\mathcal{C}$ are isomorphic is as difficult as for general graphs, up to a polynomial time transformation. For a precise definition, for instance, see the monograph [11]. The next result is not only of theoretical interest. To the contrary, for practical applications it can be read as: finding the symmetries of an integer linear program via reducing to automorphisms of suitable (labeled) graphs, is the right thing to do.

Theorem 5 *The classes of ILP graphs and reduced ILP graphs are both graph isomorphism complete.*

Proof We only prove that the class of reduced ILP graphs is graph isomorphism complete. It is known that the class of bipartite graphs is graph isomorphism complete. Hence it suffices to encode an arbitrary bipartite graph as a reduced ILP graph, which is not too large.

Let $G = (V, E)$ be an undirected bipartite graph with $m + n$ nodes $V = U \cup W = \{u_1, \dots, u_m\} \cup \{w_1, \dots, w_n\}$. As our matrix $A_G = (a_{ij}) \in \mathbb{R}^{m \times n}$ we take the bipartite adjacency matrix of G, that is,

$$a_{ij} = \begin{cases} 1 & \text{if } \{u_i, w_j\} \in E \\ 0 & \text{otherwise.} \end{cases}$$

Let G' be another bipartite graph. Then it is easy to see that the reduced ILP graph of $\mathrm{ILP}(A_G, \mathbb{1}, \mathbb{1})$ is isomorphic to the reduced ILP graph of $\mathrm{ILP}(A_{G'}, \mathbb{1}, \mathbb{1})$ if and only if G is isomorphic to G'. □

Remark 13 Margot [16, §3] observes that, in general, deciding if the group of symmetries of an ILP is the full symmetric group is an NP-complete problem. This does not conflict with our Theorem 5: Margot argues via NP-completeness of the ILP feasibility problem, while we always assume that the ILPs that we consider are feasible.

Remark 14 Rehn investigates arbitrary automorphisms of the integer lattice $\mathbb{Z}^n$ in the context of polyhedral geometry [23]. In particular, his modification of a backtracking algorithm of Plesken and Souvignier [20] allows to obtain matrix generators of the group of symmetries. For practical applications this should be superior to our approach via graph automorphisms if the number m of constraints is much larger than the dimension n.

7 Hypertruncated cubes

In this section we will construct a specific class of highly symmetric convex polytopes among which one can find examples of rather high Gomory-Chvátal rank. The motivation for this construction is rooted in the systematic study of Gomory-Chvátal cuts and cutting-plane proof systems. Pokutta and Stauffer [21] propose a new method for computing lower bounds on the Gomory-Chvátal rank for polytopes contained in the 0/1-cube, and the polytopes constructed here provide examples which asymptotically almost attain the bounds obtained. The subsequent section on computational experiments also contains results about these polytopes.

Our construction starts out with the unit cube $C = [0, 1]^n$. Intersecting C with the hyperplane defined by $\sum x_i = r$ for $r \in \{2, 3, \dots, n-1\}$ gives the hypersimplex $\Delta(r, n)$ which already appeared in Example 4. Here we are interested in the (r, n)-*truncated cube* $C' = \{x \in [0, 1]^n \mid \sum x_i \leq r\}$. Now the $(r, n; \lambda)$-*hypertruncated cube* is defined as

$$C'' = \operatorname{conv}(C' \cup \lambda \mathbb{1}) \quad \text{for } \lambda > r/n. \tag{11}$$

Notice that the full group $\operatorname{Sym}(n)$ acts on the cube C as well as on the truncated cube C' as well as on the hypertruncated cube C''. Hence our algorithms above can be applied. Our next goal is to describe the vertices and the facets of C''.

Proposition 7 *Let $n \geq 2, r \in \{2, 3, \dots, n-1\}$, and $r/n < \lambda < 1$. The vertices of the $(r, n; \lambda)$-hypertruncated cube C'' are*

$$e_S \quad \textit{for all } S \subset [n] \textit{ with } \#S \leq r \qquad \textit{and } \lambda\mathbb{1}.$$

Proof The points e_S, for $S \subset [n]$ and $\#S \leq r$, are the vertices of the (r, n)-truncated cube C'. They are also vertices of C''. Since $n\lambda$ exceeds r, the hyperplane $\sum x_i = n\lambda$ does not separate C'', and its intersection with C'' only contains the point $\lambda\mathbb{1}$. Hence the latter is a vertex, too. Looking at the defining equation (11) shows that there cannot be any other vertices. □

Of course, the vertices determine the facets completely. In this case, it is particularly easy to read off the facets of C'' by looking at the facets of C' and analyzing what changes in case the point $\lambda\mathbb{1}$ is added as a generator. This proves the claim in [21, Remark 3.3].

Proposition 8 *Let $n \geq 2, r \in \{2, 3, \dots, n-1\}$, and $\lambda > r/n$. The facets of the $(r, n; \lambda)$-hypertruncated cube C'' are*

$$x_i \geq 0, \quad x_i \leq 1 \tag{12}$$

$$\left(1 - n + \frac{r}{\lambda}\right) x_i + \sum_{k \neq i} x_k \ \geq \ r \tag{13}$$

$$\big(1 - r + \lambda(n-1)\big) x_i + (1 - \lambda) \sum_{k \neq i} x_k \leq \lambda(n - r) \tag{14}$$

for $i \in [n]$. In particular, C'' has precisely $4n$ facets.

Proof The facets of type (12) are the facets of the unit cube C. Together with the truncating inequality $\sum x_i \leq r$ they also form the facets of the truncated cube C'. The remaining facets of C'' are the facets through the vertex $\lambda\mathbb{1}$. Each of them is the convex hull of $\lambda\mathbb{1}$ and a ridge of C' contained in the truncating facet. A *ridge* is a face of codimension 2, that is, a facet of a facet. As pointed out above the truncating facet is the hypersimplex $\Delta(r, n)$. Its facets arise from the intersection with the cube facets. A hypersimplex facet of type $\{x \in \Delta(r, n) \mid x_i = 0\}$ is a *deletion facet*, and a hypersimplex facet of type $\{x \in \Delta(r, n) \mid x_i = 1\}$ is a *contraction facet*. The $n - 1$ points re_k for $k \neq i$ span an $(n - 2)$-dimensional affine subspace $\mathcal{A}$ containing the ith deletion facet. However, these points are not contained in $\Delta(r, n)$. Looking for an affine hyperplane containing $\mathcal{A}$ and $\lambda\mathbb{1}$ results in a rank-1 system of linear equations. This way we obtain the n linear inequalities of type (13). Similarly, the affine span of a contraction ridge is generated by the $n - 1$ points $e_i + (r - 1)e_k$ for $k \neq i$. Via the same approach we arrive at the n linear inequalities of type (14). □

Remark 15 Pokutta and Stauffer [21] show that the Gomory-Chvátal ranks of the $(r, n; \lambda)$-hypertruncated cubes for $r = \lfloor n/e \rfloor$, where $e = 2.7172\ldots$ is Euler's constant, and $\lambda = (m - 1)/m$ approach $n/e - o(1)$ as $m \in \mathbb{N}$ goes to infinity. In our experiments below we look at the case $r = \lfloor n/e \rfloor$ and $\lambda = 1/2$, that is, $m = 2$. Further experiments (not documented here) seem to suggest that the parameter m does not have a strong impact on the computation; this is why we restrict our attention to a fixed choice of $m = 2$.

8 Computational results

The following experiments were carried out on an Intel(R) Core(TM) i7 920 2.67 GHz machine, with 12 GB of main memory, running Ubuntu 10.04 (Lucid Lynx). The performance of each core is estimated at 5346.16 bogomips each. All our tests were run single-threaded.

The goal of the experiments is to compare the performances of a conventional branch-and-cut approach (with automated symmetry detection) and the Core-Point-Algorithm B on highly symmetric integer linear programs. For the test of the conventional branch-and-cut method we used `CPLEX`, Version 12.1.0, while the Core-Point-Algorithm was implemented and tested in `polymake`, Version 2.9.9 [8]. As a major difference `polymake` employs exact rational arithmetic (via `GMP`), while `CPLEX` uses floating-point arithmetic. It should be stressed that `CPLEX` can detect if the symmetry group of an integer linear program contains the full symmetric group acting on the standard basis of $\mathbb{R}^n$, and this is exploited in its algorithms. For input in this category (which includes all our examples below), it is thus quite a challenge to beat `CPLEX`.

8.1 Hypertruncated cubes

We tested our algorithms on the $(\lfloor n/e \rfloor, n; 1/2)$-hypertruncated cubes; see Remark 15. In this case we have only $4n$ linear inequalities from Proposition 8 as input. Each coefficient is small, and computationally accuracy (for floating-point computations) or coefficient growth (for exact arithmetic) is not an issue here. This benign input can be dealt with easily up to high dimensions. Table 1 lists the timings for `CPLEX`'

Table 1 Hypertruncated cubes

d	CPLEX		Polymake	
	Time LP (s)	Time IP (s)	Time LP (s)	Time IP (s)
100	0.00	0.07	0.01	0.08
200	0.07	0.29	0.02	0.57
300	0.19	0.73	0.03	1.88
400	0.41	1.58	0.06	4.26
500	0.90	2.99	0.10	8.39
600	1.52	4.80	0.14	14.30
700	2.21	7.01	0.18	22.29
800	3.44	11.59	0.24	32.96
900	5.17	16.37	0.31	47.11
1000	6.77	21.66	0.38	65.11
1100	8.25	26.55	0.47	85.52
1200	11.75	35.47	0.55	111.59
1300	14.05	45.63	0.65	142.76
1400	18.96	57.23	0.74	175.23
1500	23.42	73.50	0.85	217.19
1600	28.11	78.13	0.98	263.21
1700	32.07	97.23	1.11	315.38
1800	41.82	128.88	1.25	374.63
1900	44.68	137.22	1.40	444.48
2000	50.39	154.35	1.54	511.59

Branch-and-Cut and polymake's Core Point Algorithm. The timings required to obtain the solution of the linear relaxation are given separately for both systems.

The fact that polymake takes more time is due to the overhead induced by the GMP exact rational arithmetic. Since coefficient growth does not occur the overhead versus floating-point arithmetic can be estimated to be constant. Hence the roughly quadratic overhead (in dependence of n) versus the CPLEX result is a consequence of the total algorithmic complexity of $O(mn^2)$ from Corollary 3. Altogether both solvers behave pretty well for these kinds of examples.

An industry strength solver as CPLEX comes with a number of bolts and whistles which allow to tune its behavior in many ways. For the hypertruncated cubes this does not play any role. Since no parallel implementation of the Core Point Algorithm is available (yet) we set the number of CPLEX' parallel threads to one.

8.2 Wild input

One way to produce symmetric input for (integer) linear optimization algorithms is by brute force: one can take any system $Ax \leq b$ of linear inequalities and let the full group $\mathrm{Sym}(n)$ act. This way each original inequality may give up to $n!$ inequalities in the resulting *symmetrized* system. The symmetrized system of inequalities is $\mathrm{Sym}(n)$-

invariant by construction. In order to produce input to our algorithms which is less well behaved than the hypertruncated cubes studied above we will apply this procedure to a special class of polytopes, which can be considered "wild". We aim at symmetric polytopes with many facets whose coordinates are not so nice, but still somewhat under control.

The first building block of our construction is the regular hexagon H whose vertices are at distance $56/6$ from the origin, that is,

$$H = \operatorname{conv}\left\{ \frac{56}{6} e^{k\pi i/6} \;\middle|\; k = 0, 1, \ldots, 5 \right\}.$$

Notice that only in the formula above the letter 'i' denotes the imaginary unit, and we identify the complex numbers with $\mathbb{R}^2$. The coordinates of H are irrational; however, the subsequent steps in the construction are chosen such that we will arrive at a rational polytope in the end. The second item is the regular cross polytope scaled by $73/10$, that is,

$$C(d) = \operatorname{conv}\left\{ \pm\frac{73}{10} e_i \;\middle|\; i \in [d] \right\}.$$

Finally, we consider the *join* $P * Q$ of two polytopes $P \subset \mathbb{R}^\delta$ and $Q \subset \mathbb{R}^\varepsilon$, which is defined as

$$\begin{aligned} P * Q = \operatorname{conv}\big(&\left\{ (x, 0, 1) \in \mathbb{R}^\delta \times \mathbb{R}^\varepsilon \times \mathbb{R} \;\middle|\; x \in P \right\} \\ &\cup \left\{ (0, y, -1) \in \mathbb{R}^\delta \times \mathbb{R}^\varepsilon \times \mathbb{R} \;\middle|\; y \in Q \right\} \big). \end{aligned}$$

If P and Q are full-dimensional polytopes with μ and ν vertices, respectively, the join $P * Q$ has dimension $\delta + \varepsilon + 1$ and $\mu + \nu$ vertices. For the combinatorics of $P * Q$ the exact values for the $(\delta + \varepsilon + 1)$st coordinate are inessential, as long as they are distinct. We now replace the "-1" for the second factor by $-11/12$ to obtain the *distorted join*

$$\begin{aligned} J(d) = \operatorname{conv}\Bigg(&\left\{ (x, 0, 1) \in \mathbb{R}^2 \times \mathbb{R}^d \times \mathbb{R} \;\middle|\; x \in H \right\} \\ &\cup \left\{ \left(0, y, -\frac{11}{12}\right) \in \mathbb{R}^2 \times \mathbb{R}^d \times \mathbb{R} \;\middle|\; y \in C(d) \right\} \Bigg) \end{aligned}$$

of the hexagon H with the cross polytope $C(d)$. This polytope is further modified in two steps: First we perturb by rounding the (rational and irrational) coordinates to three decimal places and treating these as exact rational numbers. Since the polytopes H, $C(d)$, and $J(d)$ are simplicial this perturbation does not change the combinatorial types. Secondly, we symmetrize the polytope by letting the group $\operatorname{Sym}(d + 3)$ act on the facets of the perturbed polytope. The resulting inequalities form the input of our second class of experiments.

The parameters $56/6$, $73/10$, and $11/12$ which occur in the construction are chosen, more or less, at random. They do not have a specific meaning. We refrain from

Table 2 Symmetrized distorted joins of a hexagon with cross-polytopes

d	CPLEX		Polymake	
	Time LP (s)	Time IP (s)	Time LP (s)	Time IP (s)
3	0.00	0.01	0.00	0.00
4	0.00	0.06	0.01	0.00
5	0.00	0.17	0.01	0.02
6	0.05	0.74	0.04	0.04
7	0.13	2.71	0.09	0.13
8	0.62	10.15	0.24	0.38
9	2.08	42.06	0.69	1.03
10	8.02	135.51	1.86	2.89

further investigating these symmetrized distorted joins and the geometry of lattice points inside. This would be tedious and at the same time irrelevant for our purposes.

The interesting fact is that we get symmetric polytopes which are somewhat complicated, because they have lots of inequalities: for instance, yielding 885,768 inequalities for $d = 10$. As a consequence CPLEX cannot deal with these examples in a fully automated way. The best parameter settings that we found were

Parallel thread count	1
Presolve indicator	No
Feas. pump heuristic	-1
RINS heuristic	-1
MIP optimization emph.	2

But even with these adjustments our implementation outperforms CPLEX by a large margin; see Table 2. This holds in spite of the fact that polymake computes with exact rational numbers throughout.

Acknowledgments We are indebted to Tobias Achterberg, Max Horn, Leo Liberti, Marc Pfetsch, Sebastian Pokutta, and Achill Schürmann for valuable discussions on various aspects of this paper. Moreover, we would like to thank an anonymous referee for pointing out the reference [22].

References

1. Berthold, T., Pfetsch, M.E. : Detecting orbitopal symmetries. In: Fleischmann, B., Borgwardt, K.H., Klein, R., Tuma, A. (eds.) Operations Research Proceedings 2008, pp. 433–438. Springer, Berlin (2009)
2. Bixby, R., Wagner, D.K.: A note on detecting simple redundancies in linear systems. OR Lett. **6**(1), 15–17 (1987)
3. Bödi, R., Grundhöfer, T., Herr, K.: Symmetries of linear programs. Note Mat. **30**(1), 129–132 (2010)
4. Bremner, D., Dutour Sikirić, M., Schürmann, A.: Polyhedral representation conversion up to symmetries. In: Polyhedral Computation, CRM Proceedings. Lecture Notes, vol. 48, pp. 45–71. American Mathematical Society, Providence (2009)
5. Cameron, P.J.: Finite permutation groups and finite simple groups. Bull. Lond. Math. Soc. **13**, 1–22 (1981)

6. Friedman, E.J.: Fundamental domains for integer programs with symmetries. In: Zu, Y., Zhu, B., Dress, A. (eds.) COCOA 2007, LNCS, vol. 4616, pp. 146–153. Springer, Berlin (2007)
7. Gatermann, K., Parrilo, P.A.: Symmetry groups, semidefinite programs, and sums of squares. J. Pure Appl. Algebra **192**(1–3), 95–128 (2004). doi:10.1016/j.jpaa.2003.12.011
8. Gawrilow, E., Joswig, M.: polymake: a framework for analyzing convex polytopes. In: Polytopes—combinatorics and computation (Oberwolfach, 1997), DMV Sem., vol. 29, pp. 43–73. Birkhäuser, Basel (2000)
9. Kaibel, V., Pfetsch, M.: Packing and partitioning orbitopes. Math. Program. **114**(1, Ser. A), 1–36 (2008). doi:10.1007/s10107-006-0081-5
10. Kaibel, V., Schwartz, A.: On the complexity of polytope isomorphism problems. Graphs Combin. **19**(2), 215–230 (2003)
11. Köbler, J., Schöning, U., Torán, J.: The Graph Isomorphism Problem: Its Structural Complexity. Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston (1993)
12. Lenstra, H.W. Jr.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983). doi:10.1287/moor.8.4.538
13. Liberti, L.: Reformulations in mathematical programming: automatic symmetry detection and exploitation. Math. Program. 1–32 (2010). doi:10.1007/s10107-010-0351-0
14. Margot, F.: Pruning by isomorphism in branch-and-cut. Math. Program. **94**(1, Ser. A), 71–90 (2002). doi:10.1007/s10107-002-0358-2
15. Margot, F.: Exploiting orbits in symmetric ILP. Math. Program. **98**(1–3, Ser. B), 3–21 (2003). doi:10.1007/s10107-003-0394-6. Integer programming (Pittsburgh, PA, 2002)
16. Margot, F.: Symmetry in integer linear programming. In: Jünger, M., Liebling, T., Naddef, D., Nemhauser, G.L., Pulleyblank, W., Reinelt, G., Rinaldi, G., Wolsey, L. (eds.) 50 Years of Integer Programming 1958–2008, chap. 17, pp. 647–681. Springer, Berlin (2010)
17. McKay, B.D.: Nauty (2005). http://cs.anu.edu.au/people/bdm/nauty
18. Miller, G.A.: Multiply transitive substitution groups. Trans. Am. Math. Soc. **28**(2), 339–345 (1926). doi:10.2307/1989119
19. Ostrowski, J., Linderoth, J., Rossi, F., Smriglio, S.: Orbital branching. Math. Program. **126**(1, Ser. A), 147–148 (2011). doi:10.1007/s10107-009-0273-x
20. Plesken, W., Souvignier, B.: Computing isometries of lattices. J. Symb. Comput. **24**(3–4), 327–334 (1997). doi:10.1006/jsco.1996.0130. Computational algebra and number theory (London, 1993)
21. Pokutta, S., Stauffer, G.: Lower bounds for the chvátal-gomory rank in the 0/1 cube. Oper. Res. Lett. **39**(3), 200–203 (2011). doi:10.1016/j.orl.2011.03.001; http://www.sciencedirect.com/science/article/pii/S0167637711000289
22. Puget, J.F.: Automatic detection of variable and value symmetries. In: van Beek, P. (ed.) Principles and Practice of Constraint Programming—CP 2005, Lecture Notes in Computer Science, vol. 3709, pp. 475–489. Springer, Berlin (2005). http://dx.doi.org/10.1007/11564751_36
23. Rehn, T.: Polyhedral Description Conversion Up to Symmetries. Master's thesis, Otto-von-Guericke Universität Magdeburg (2010). http://www.math.uni-rostock.de/~rehn/docs/diploma-thesis-ma-rehn.pdf
24. Rehn, T., Schürmann, A.: A C++ tool for the work with symmetric polyhedra. Preliminary version 0.1.1. http://fma2.math.uni-magdeburg.de/~latgeo/sympol/sympol.html
25. Rossmann, W.: An introduction through linear groups. In: Lie Groups, Oxford Graduate Texts in Mathematics, vol. 5. Oxford University Press, Oxford (2002)
26. Webb, P.: Finite group representations for the pure mathematician. http://www.math.umn.edu/~webb/RepBook/index.html
27. Ziegler, G.M.: Lectures on Polytopes, Graduate Texts in Mathematics, vol. 152. Springer, New York (1995)