

OLGA HEISMANN      ACHIM HILDENBRANDT      FRANCESCO SILVESTRI  
GERHARD REINELT      RALF BORNDÖRFER

## **HUHFA: A Framework for Facet Classification**

Herausgegeben vom  
Konrad-Zuse-Zentrum für Informationstechnik Berlin  
Takustraße 7  
D-14195 Berlin-Dahlem

Telefon: 030-84185-0  
Telefax: 030-84185-125

e-mail: [bibliothek@zib.de](mailto:bibliothek@zib.de)  
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064  
ZIB-Report (Internet) ISSN 2192-7782

# HUHFA: A Framework for Facet Classification

Olga Heismann\*    Achim Hildenbrandt\*\*    Francesco Silvestri\*\*  
Gerhard Reinelt\*\*    Ralf Borndörfer\*

## Abstract

Usually complete linear descriptions of polytopes consist of an enormous number of facet-defining inequalities already for very small problem sizes. In this paper, we describe a method for dividing the inequalities into equivalence classes without resorting to a normal form. Within each class, facets are related by certain symmetries and it is sufficient to list one representative of each class to give a complete picture of the structural properties of a polytope. We propose an algorithm for the classification and illustrate its efficiency on a broad range of combinatorial optimization problems including the Traveling Salesman and the Linear Ordering Problem.

## 1 Introduction

The polyhedral approach to combinatorial optimization problems studies the structure of their associated polytopes. One way is to compute complete linear descriptions of small polytopes in order to generalize the equations and inequalities. “Small polytopes” might actually not look so small at first sight: There is often a huge number of facet-defining inequalities already for very small problem sizes.

However, there are also often many symmetries implied by the combinatorial structure of the problem which can be used to classify the facets. These symmetries act on the feasible solutions and naturally form a group. In their representation as maps on the variable values they can be extended to symmetries acting on the polytope, and one can prove that they map vertices of the polytope to vertices of the polytope, and facets to facets. We say that those facet-defining inequalities which are similar in the sense that they can be transformed onto each other by some symmetry belong to one class.

Understanding all the facet-defining inequalities of a combinatorial optimization problem polytope then reduces to understanding one facet from each class.

To do this classification, one applies the symmetries to the facet-defining inequalities and then checks whether any two facets can be transformed into each other and hence belong to

---

\*Zuse Institute Berlin, Takustraße 7, 14195 Berlin, Germany, {borndoerfer, heismann}@zib.de. Supported by the DFG Research Center MATHEON “Mathematics for key technologies”.

\*\*Ruprecht-Karls-Universität Heidelberg, INF 368, 69120 Heidelberg, Germany, {achim.hildenbrandt, gerhard.reinelt}@informatik.uni-heidelberg.de, f.silvestri@stud.uni-heidelberg.de.

the same class. Often, this check is not so easy as two linear expressions describing the same facet might differ by the sum of multiples of several equalities from the problem description.

The check can be accomplished by defining a so-called normal form for the representation of inequalities—inequalities which have the same normal form describe the same facet. To this end, problem-specific normal forms were developed for some extensively studied combinatorial optimization problems. For the Traveling Salesman Problem, every facet-defining inequality can be efficiently transformed to the so-called tight triangular normal form [Naddef and Rinaldi, 1993]. For an example of a normal form for the Linear Ordering Problem, see [Reinelt, 1985]. In general, the representation of facet-defining inequalities in the orthogonal complement of the linear subspace spanned by the equations can be of course used as a normal form for the facets of a polytope. However, this needs techniques from linear algebra and can therefore raise numerical issues. Unfortunately, normal forms that can be described combinatorially are often not known. Hence, having a method that can be applied to every combinatorial optimization problem and relies solely on the combinatorial structure of the polytope is desirable.

Indeed, in this paper we propose a novel technique for classifying facets without using normal forms. The main idea is to identify every facet-defining inequality with the vertices of the polytope which satisfy it with equality. With this method, complete descriptions of polytopes computed by a software like PORTA [Christof and Loebel, 2008] (or a similar package) can be analyzed to divide the facets into equivalence classes according to groups generated by given symmetry mappings. It works regardless of whether the polytope is full-dimensional or not.

Facet classification methods without normal forms are also used in the Software SymPol [Rehn and Schürmann, 2010] for polyhedral representation transformations. To the best of our knowledge, their method relies on geometric scalar product invariants and algebraic invariants using polynomial rings as described in [Rehn, 2010, Section 3.2.2]. The invariant proposed in this paper is much easier to compute and relies only on the vertex-facet incidence structure of the polytope.

This paper is organized as follows. In Section 2 we present our approach for the classification of facet-defining inequalities. Section 3 gives some examples for symmetries, and in Section 4 we make a few comments about extensions of our theory for the classification of equations, which can be present in linear descriptions. Finally, we describe the implemented algorithm in Section 5 and give some computational results. The paper closes with a conclusion in Section 6.

## 2 Equivalence of Facets

A *polyhedron*  $P$  is the solution set of a finite system of linear equations and inequalities, i. e.,  $P$  can be described as  $P = \{x \in \mathbb{R}^n : Ax \leq b, Cx = d\}$  for some  $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, C \in \mathbb{R}^{l \times n}, d \in \mathbb{R}^l$ . A bounded polyhedron is called a *polytope*. The *dimension*  $\dim(P)$  of  $P$  is the cardinality of a maximum affinely independent subset minus one.

In the following we will only consider polytopes as feasible solutions of combinatorial

optimization problems are usually bounded. Furthermore, we will not give the explicit dimensions of real vectors or matrices whenever they are clear from the context.

An inequality  $a^T x \leq b$  is called a *valid inequality for  $P$*  if it is satisfied by all elements of  $P$ . For a valid inequality, the set  $F = \{x : a^T x = b\} \cap P$  is called a *face of  $P$* . A face  $F$  is called a *facet of  $P$*  if  $\dim F = \dim P - 1$ . The inequality inducing the facet is called *facet-defining*. A face consisting of a single element is called a *vertex*.

A linear representation  $\{x : Ax \leq b, Cx = d\}$  of a polytope is called an  $\mathcal{H}$ -*representation of  $P$* .

The *convex hull*  $\text{conv}(X)$  of a finite set  $X = \{x_1, x_2, \dots, x_t\}$  is the set of all vectors  $z$  which can be written as a (convex) combination  $z = \sum_{i=1}^t \lambda_i x_i$ ,  $0 \leq \lambda_i \leq 1$ ,  $i = 1, \dots, t$ .

It is a fundamental theorem in polyhedral theory (see, e. g., [Weyl, 1934]) that a polytope  $P$  can also be described as the convex hull of its vertices  $\text{vert}(P)$ . For  $V = \text{vert}(P)$  we call  $\text{conv}(V)$  the  $\mathcal{V}$ -*representation of  $P$* .

The following definition of symmetries will allow us to view facet classes as equivalence classes.

**Definition 2.1.** Let  $s : x \mapsto Mx + r$  be a bijection on  $\mathbb{R}^n$  with some (nonsingular) matrix  $M$  and a vector  $r$ . The faces  $F_1$  and  $F_2$  of a polytope  $P$  are *equivalent with respect to  $s$*  if  $s(\text{vert}(F_1)) = \text{vert}(F_2)$ . If  $S$  is a set of bijections, then  $F_1$  and  $F_2$  are *equivalent with respect to  $S$*  if they are equivalent with respect to  $s$  for some  $s \in S$ . A bijection  $s$  is said to be a *symmetry for  $P$*  if  $s(\text{vert}(P)) = \text{vert}(P)$ .

The following two lemmas establish useful properties of a symmetry.

**Lemma 2.2.** Let  $s : x \mapsto Mx + r$  be a bijection on  $\mathbb{R}^n$  with some (nonsingular) matrix  $M$  and a vector  $r$ . Then  $s^{-1} : x \mapsto M^{-1}x - M^{-1}r$ .

*Proof.* Since  $s$  is bijective, an inverse exists. We compute that  $t(s(x)) = x$  for  $t(x) = M^{-1}x - M^{-1}r$ :

$$\begin{aligned} t(s(x)) &= M^{-1}(Mx + r) - M^{-1}r \\ &= x + M^{-1}r - M^{-1}r \\ &= x. \end{aligned}$$

□

**Lemma 2.3.** Let  $P$  be a polytope and  $s : x \mapsto Mx + r$  be a symmetry for  $P$  with some (nonsingular) matrix  $M$  and a vector  $r$ . Then  $s(P) = P$ .

*Proof.* Consider the  $\mathcal{V}$ -representation of  $P$  with  $V := \text{vert}(P)$ :

$$x = \sum_{v \in V} \lambda_v \cdot v, \quad \sum_{v \in V} \lambda_v = 1, \quad 0 \leq \lambda_v \leq 1 \quad \forall v \in V$$

for every  $x \in P$ . Then,

$$\begin{aligned}
s(x) &= s\left(\sum_{v \in V} \lambda_v \cdot v\right) \\
&= M \cdot \left(\sum_{v \in V} \lambda_v \cdot v\right) + r \\
&= M \cdot \left(\sum_{v \in V} \lambda_v \cdot v\right) + r \cdot \left(\sum_{v \in V} \lambda_v\right) \\
&= \sum_{v \in V} \lambda_v \cdot (Mv + r) \\
&= \sum_{v \in V} \lambda_{s^{-1}(v)} v \in \text{conv}(V) = P.
\end{aligned}$$

Thus,  $s(x) \in P$  for every  $x \in P$ .

Since  $s^{-1}(x) = M^{-1}x - M^{-1}r$  is a symmetry, too, the argument from above proves also that  $s^{-1}(x) \in P$  for every  $x \in P$ .  $s(x) \in P$  and  $s^{-1}(x) \in P$  for every  $x \in P$  implies that  $s(P) = P$ .  $\square$

In our combinatorial understanding, a symmetry acts on the feasible solutions of a combinatorial optimization problem. This might seem not to match Definition 2.1 where a symmetry acts only on those feasible solutions which are vertices of the corresponding polytope. The vertices might be only a subset of the feasible solutions. However, this is not a restriction as feasible solutions that are vertices have to be mapped to feasible solutions that are vertices, too, by a symmetry in the combinatorial understanding. This is proven in the following lemma.

**Lemma 2.4.** *Let  $s : x \mapsto Mx + r$  be a bijection on  $\mathbb{R}^n$  with some (nonsingular) matrix  $M$  and a vector  $r$ , and let  $P$  be a polytope. If  $s(S) = S$  for some set  $\text{vert}(P) \subseteq S \subseteq P$ , then  $s(\text{vert}(P)) = \text{vert}(P)$ .*

*Proof.* We have to prove that for every  $v \in \text{vert}(P)$ ,  $s(v) \in \text{vert}(P)$ .

Since  $v$  is a face, a valid inequality  $a^T x \leq b$  exists for  $P$  such that  $\{v\} = \{x : a^T x = b\} \cap P$ . The inequality  $a^T s^{-1}(x) \leq b \Leftrightarrow (a^T M^{-1})x \leq aM^{-1}r + b$  is fulfilled by  $s(v)$  with equality, and it is valid for  $s(S)$ , because  $S \subseteq P$  and  $a^T x \leq b$  is valid for  $P$ . On the other hand, for every  $s(v')$  that fulfills  $(a^T M^{-1})x \leq aM^{-1}r + b$  with equality,  $a^T v' = b$ , so that  $v$  is the only point with this property. Thus,  $\{s(v)\} = \{x : (a^T M^{-1})x = aM^{-1}r + b\} \cap P$  is a vertex of  $P$ .  $\square$

We now show that a group of such symmetries induces indeed an equivalence relation on the faces of a polytope.

**Lemma 2.5.** *Let  $P$  be a polytope and  $S$  be a set of symmetries on  $P$  forming a group with respect to the composition operator  $(\circ)$ . Then equivalence with respect to  $S$  defines an equivalence relation on the faces of  $P$ .*

*Proof.* Reflexivity holds because  $S$  contains the identity. Transitivity follows from the closure of  $S$  and the fact that if  $\text{vert}(F_2) = s_1(\text{vert}(F_1))$  and  $\text{vert}(F_3) = s_2(\text{vert}(F_2))$ , then  $\text{vert}(F_3) = (s_2 \circ s_1)(\text{vert}(F_1))$ . Finally, symmetry follows from the existence of inverse elements in  $S$ . If  $\text{vert}(F_2) = s(\text{vert}(F_1))$ , then  $\text{vert}(F_1) = s^{-1}(\text{vert}(F_2))$ .  $\square$

**Theorem 2.6.** *Let  $F$  be a facet of the polytope  $P$  and  $s : x \mapsto Mx + r$  a symmetry for  $P$ . If  $a^T x \leq b$  defines the facet  $F$ , then  $(a^T M^{-1})x \leq b + a^T M^{-1}r$  defines some facet  $F'$  of  $P$  as well. Further,  $s(F) = F'$  holds.*

*Proof.* Since  $s$  is bijective and by Lemma 2.3  $s^{-1}(x) \in P$  for every  $x \in P$ , the inequality  $a^T s^{-1}(x) \leq b$  holds for every  $x \in P$  as well. Written explicitly, we have  $a^T (M^{-1}x - M^{-1}r) = a^T s^{-1}(x) \leq b$  and therefore the inequality  $a^T M^{-1}x \leq b + a^T M^{-1}r$  is valid for  $P$ , too.

To complete the proof we will show that  $\dim(F') = \dim(F)$  where

$$F' := \{x : a^T M^{-1}x = b + a^T M^{-1}r\} \cap P.$$

Because  $s$  is a bijection we get

$$\begin{aligned} a^T x = b &\Leftrightarrow a^T s^{-1}(s(x)) = b \\ &\Leftrightarrow a^T (M^{-1}s(x) - M^{-1}r) = b \\ &\Leftrightarrow (a^T M^{-1})s(x) = b + a^T M^{-1}r \\ &\Leftrightarrow s(x) \in \{x : a^T M^{-1}x = b + a^T M^{-1}r\}. \end{aligned}$$

By Lemma 2.3 for  $s$  and  $s^{-1}$ ,  $x \in P \Leftrightarrow s(x) \in P$ . Using the equivalence shown above, we get

$$\begin{aligned} x \in F &\Leftrightarrow x \in P \wedge a^T x = b \\ &\Leftrightarrow s(x) \in P \wedge a^T M^{-1}s(x) = b + a^T M^{-1}r \\ &\Leftrightarrow s(x) \in F'. \end{aligned}$$

Hence,  $s(F) = F'$ . Since  $s$  is an affine bijective map, it is dimension-preserving and the proof is completed.  $\square$

### 3 Examples and Groups of Bijections

In this section, we investigate some interesting groups of symmetries. We begin with examples of symmetries in two well-known combinatorial optimization problems. Afterwards, we prove possibilities to generate symmetry groups, and illustrate them using the hypercube as an example.

**Example 3.1.** Given the complete directed graph  $D_n = (V_n, A_n)$  on  $n$  nodes, a *Hamiltonian cycle (tour)* in  $D_n$  is a directed cycle that contains all nodes. For given arc weights the well-known *Asymmetric Traveling Salesman Problem (ATSP)* consists of finding a Hamiltonian cycle (tour) in  $D_n$  of minimum total length.

If a tour and a permutation  $\sigma$  of the nodes  $V_n$  are given, then the replacement of every arc  $(i, j)$  by  $(\sigma(i), \sigma(j))$  yields again a tour. Furthermore, changing the direction of all arcs, i. e., replacing each arc  $(i, j)$  by  $(j, i)$ , also gives a feasible tour. Thus, these two operations can be viewed as symmetries on the set of tours.

Consider the standard linear characterization of tours with binary variables  $x_{ij}$  where  $x_{ij} = 1$  if arc  $(i, j)$  is in the tour, and  $x_{ij} = 0$ , otherwise.

$$\sum_{i=1, i \neq j}^n x_{ij} = 1, \text{ for all } j \in V_n, \quad (\text{TSP i})$$

$$\sum_{j=1, j \neq i}^n x_{ij} = 1, \text{ for all } i \in V_n, \quad (\text{TSP ii})$$

$$\sum_{i \in S} \sum_{j \in S, i \neq j} x_{ij} \leq |S| - 1, \text{ for all } S \subset V_n, 2 \leq |S| \leq n - 1, \quad (\text{TSP iii})$$

$$x_{ij} \in \{0, 1\}, \text{ for all } (i, j) \in A_n. \quad (\text{TSP iv})$$

The *Asymmetric Traveling Salesman Polytope*  $P_{ATSP}^n$  is defined as the convex hull of all feasible 0/1 vectors of this characterization.

If  $\sigma$  is a permutation of  $V_n$ , then the map  $s^\sigma$  with  $(s^\sigma(x))_{ij} = x_{\sigma^{-1}(i)\sigma^{-1}(j)}$  is an affine map. By applying  $s^\sigma$  to a feasible solution  $x$  we obtain the feasible solution  $x'$  where

$$x'_{ij} := \begin{cases} 1 & \text{if } x_{\sigma^{-1}(i)\sigma^{-1}(j)} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The other symmetry, which reverses arcs, can be represented as the affine map  $r$  with  $(r(x))_{ij} = x_{ji}$  converting a tour  $x$  to the tour  $x''$ :

$$x''_{ij} := \begin{cases} 1 & \text{if } x_{ji} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Both maps  $s^\sigma$  and  $r$  are bijective. Further, they map the vertices of  $P_{ATSP}^n$  onto themselves, so they are symmetries for  $P_{ATSP}^n$ .

E. g., consider the inequality  $x_{12} + x_{13} + x_{14} \leq 1$  which is facet-defining for  $P_{ATSP}^4$ , and let  $\sigma$  be the permutation interchanging nodes 1 and 2. If we apply  $s^\sigma$  and  $r$  we obtain the facet-defining inequality  $x_{12} + x_{32} + x_{42} \leq 1$  that belongs to the same equivalence class.

**Example 3.2.** Let  $D_n = (V_n, A_n)$  be the complete directed graph on  $n$  nodes. A *tournament* in  $D_n$  is a subset of  $A_n$  which contains for every pair  $(i, j)$  of nodes exactly one of the arcs  $(i, j)$  or  $(j, i)$ . For given arc weights the well-known *Linear Ordering Problem (LOP)* consists of finding an acyclic tournament of maximum total weight.



With binary variables  $y_{ij}$  indicating whether arc  $(i, j)$  is in the tournament or not a linear characterization of acyclic tournaments is given by the system

$$\begin{aligned}
y_{ij} + y_{ji} &= 1, \text{ for all } 1 \leq i < j \leq n, & \text{(LOP i)} \\
y_{ij} + y_{jk} + y_{ki} &\leq 2, \text{ for all } 1 \leq i, j, k \leq n, i < j, i < k, k \neq j, & \text{(LOP ii)} \\
y_{ij} &\leq 1, \text{ for all } 1 \leq i, j \leq n, j \neq i, & \text{(LOP iii)} \\
y_{ij} &\in \mathbb{Z}, \text{ for all } (i, j) \in A_n. & \text{(LOP iv)}
\end{aligned}$$

The *Linear Ordering Polytope* is the convex hull of all feasible 0/1 vectors of this characterization.

As for the ATSP, also here node permutations and arc reversals can be used to convert acyclic tournaments into other acyclic tournaments. In addition to these symmetries there is another type given in [Bolotashvili et al., 1999]. For a node  $r \in V_n$  define the bijective map  $\phi_r : \mathbb{R}^{n(n-1)} \rightarrow \mathbb{R}^{n(n-1)}$  by

$$(\phi_r(y))_{ij} = \begin{cases} y_{ji} & \text{if } i = r \text{ or } j = r, \\ y_{ij} + y_{jr} - y_{ir} & \text{otherwise.} \end{cases}$$

This symmetry is a so-called “rotation mapping” and maps the linear ordering  $1, 2, \dots, r-1, r, r+1, \dots, n$  to the linear ordering  $r+1, \dots, n, r, 1, 2, \dots, r-1$ . However, it is different from a node permutation symmetry: The interesting fact is that with this mapping the 3-dicycle inequalities (LOP ii) and the trivial inequalities (LOP iii) belong to the same equivalence class, which can be checked by an easy calculation.

In combinatorial optimization problems, variables often model whether some object is selected for the solution. Then, a permutation symmetry on these objects implies such a symmetry on the variables as stated in the next definition.

**Definition 3.3.** Let  $\sigma$  be a permutation of  $\{1, \dots, n\}$ .  $\sigma$  induces a linear bijective map  $\bar{\sigma} : x = (x_1, \dots, x_n) \mapsto x_\sigma := (x_{\sigma(1)}, \dots, x_{\sigma(n)})$ .

Analogously, every subgroup  $S$  of the permutation group  $S_n$  defines a group  $\bar{S} = \{\bar{\sigma} : \sigma \in S\}$  of bijections. In the following will not distinguish between  $\sigma$  and  $S_n$  and their implied bijections  $\bar{\sigma}$  and  $\bar{S}_n$  in notation.

We denote by  $(i_1, i_2, \dots, i_k) \in S_n$  the permutation that maps  $i_j$  to  $i_{(j \bmod k)+1}$  for all  $j = 1, 2, \dots, k$ . For a group  $S$  and  $s_1, s_2, \dots, s_l \in S$ , we write  $\langle s_1, s_2, \dots, s_l \rangle$  for the subgroup of  $S$  that is generated by  $s_1, s_2, \dots, s_l$ .

The following definition and the next lemma allow to create further groups of symmetries on a polytope.

**Definition 3.4.** Let  $\tau : \mathbb{R} \rightarrow \mathbb{R}$  be an *involution*, i. e.,  $\tau \circ \tau = \text{id}$ . For a set  $I \subseteq \{1, \dots, n\}$ , we define the map  $s_I^\tau : \mathbb{R}^n \rightarrow \mathbb{R}^n$  by setting

$$s_I^\tau(x)_i = \begin{cases} \tau(x_i) & \text{if } i \in I, \\ x_i & \text{otherwise.} \end{cases}$$

It is not hard to see that  $s_I^\tau \circ s_I^\tau = \text{id}$ . In addition, define  $S_J^\tau = \{s_I^\tau : I \subseteq J\}$  and  $S^\tau = S_{\{1, \dots, n\}}^\tau$  when  $n$  is clear from context.

**Lemma 3.5.** *For every  $J \subseteq \{1, \dots, n\}$  and every involution  $\tau : \mathbb{R} \rightarrow \mathbb{R}$ ,  $S_J^\tau$  is a group of bijections with respect to composition.*

*Proof.* Because of  $\emptyset \subseteq J$  it follows that  $\text{id} = s_\emptyset^\tau \in S_J^\tau$ .

Existence of inverse functions is a direct consequence of  $s_I^\tau$  being an involution.

Closure follows from the fact that for every  $I_1, I_2 \subseteq J$  one has  $I_1 \Delta I_2 \subseteq J$  (where  $\Delta$  denotes symmetric difference) as well as  $s_{I_1}^\tau \circ s_{I_2}^\tau = s_{I_1 \Delta I_2}^\tau$ , which can be easily verified.  $\square$

Thus, for example,  $S_J^-$  with  $- : x \mapsto -x$  and  $J \subseteq \{1, \dots, n\}$  is a group of bijections with respect to composition and we could apply Lemma 2.5 to get an equivalence relation on a polytope for which the elements in  $S_J^-$  are symmetries.

Furthermore, we can show that compositions of certain types of bijections also form a group. They are defined as follows.

**Definition 3.6.** Let  $S$  and  $T$  be sets of bijections. The set  $S \circ T$  is the set of all possible compositions of elements in  $S$  and  $T$ , i. e.,  $S \circ T = \{s \circ t : s \in S, t \in T\}$ .

The following two lemmas state that we can compose  $S_n$  and  $S^-$  to generate new groups of bijections.

**Lemma 3.7.** *Let  $S, T$  be groups.  $S \circ T$  is a group if and only if  $S \circ T = T \circ S$ .*

*Proof. Part 1:*  $S \circ T = T \circ S \Rightarrow S \circ T$  group:

(i) Because  $\text{id} \in S, T$ , we can write  $\text{id} = (\text{id} \circ \text{id}) \in S \circ T$ .

(ii) To show the existence of inverse elements let  $(s \circ t) \in S \circ T$ , where  $s \in S$  and  $t \in T$ . The inverse of  $(s \circ t)$  is clearly given by  $(t^{-1} \circ s^{-1})$  which is in  $T \circ S = S \circ T$  again.

(iii)  $S \circ T = T \circ S$  means that for any  $(t \circ s) \in T \circ S$ , we have  $(t \circ s) = (s' \circ t')$  for some  $s' \in S, t' \in T$  and vice versa (\*). Using this rule, we can show that

$$\begin{aligned} (s_1 \circ t_1) \circ (s_2 \circ t_2) &= (s_1 \circ t_1) \circ (t_3 \circ s_3) \\ &= s_1 \circ t_1 \circ t_3 \circ s_3 \\ &= s_1 \circ ((t_1 \circ t_3) \circ s_3) \\ &= s_1 \circ (s_4 \circ t_4) \\ &= (s_1 \circ s_4) \circ t_4 \in S \circ T \end{aligned}$$

where the substitutions are all due to (\*). This shows composition.

**Part 2:**  $S \circ T$  is a group  $\Rightarrow S \circ T = T \circ S$ :

It suffices to show that for any  $s \in S, t \in T$ , we have  $(t, s) \in S \circ T$  and  $(s, t) \in T \circ S$ . First, consider  $(s^{-1} \circ t^{-1}) \in S \circ T$ . Because  $S \circ T$  is a group, it follows for the inverse that  $(t \circ s) = (s^{-1} \circ t^{-1})^{-1} \in S \circ T$ , which shows  $T \circ S \subseteq S \circ T$ .

We proceed to show that  $T \circ S$  is also a group, so swapping  $S$  and  $T$  in the argument before shows  $T \circ S \supseteq S \circ T$  and we are done.

Reflexivity is clear since  $(\text{id} \circ \text{id}) \in T \circ S$ . To see the closure, choose some  $s, s' \in S$  and  $t, t' \in T$ . By the closure of  $S \circ T$ ,

$$(s'^{-1} \circ t'^{-1}) \circ (s^{-1} \circ t^{-1}) = (s''^{-1} \circ t''^{-1})$$

holds for some  $s'' \in S, t'' \in T$ . Its inverse reads as

$$(t \circ s) \circ (t' \circ s') = (t'' \circ s'').$$

This proves the closure of  $T \circ S$ .

For  $(t \circ s) \in T \circ S$  it follows by closure that

$$(t \circ s)^{-1} = (s^{-1} \circ t^{-1}) = (\text{id} \circ s^{-1}) \circ (t^{-1} \circ \text{id}) \in T \circ S,$$

which shows the existence of inverse elements and completes the proof.  $\square$

**Lemma 3.8.** For the groups  $S \subseteq S_n$  and  $S^\tau$  we have  $S_n^\tau = S \circ S^\tau = S^\tau \circ S$ .

*Proof.* It suffices to show explicitly that  $\sigma \circ s_i^\tau = s_{\sigma(i)}^\tau \circ \sigma$  for  $\sigma \in S$  and  $s_i^\tau \in S^\tau$ . Then we can apply Lemma 3.7.

For  $x \in \mathbb{R}^n$  we have

$$(\sigma \circ s_i^\tau)(x)_i = \begin{cases} \tau(x_{\sigma(i)}) & \text{if } i \in I, \\ x_{\sigma(i)} & \text{otherwise,} \end{cases}$$

and

$$(s_{\sigma(i)}^\tau \circ \sigma)(x)_i = \begin{cases} \tau(x_{\sigma(i)}) & \text{if } \sigma(i) \in \sigma(I), \\ x_{\sigma(i)} & \text{otherwise.} \end{cases}$$

Since  $\sigma$  is a bijection,  $i \in I \Leftrightarrow \sigma(i) \in \sigma(I)$ , which shows that the two functions above are identical and thus  $S \circ S^\tau = S^\tau \circ S$ .  $\square$

**Example 3.9.** We define the involution

$$\neg : x \mapsto 1 - x.$$

If we view 0 and 1 as Boolean values,  $s^\neg$  simply negates them. By Lemma 3.5,  $S^\neg = \{s_i^\neg : I \subseteq \{1, \dots, n\}\}$  forms a group and defines equivalence classes on the faces of any polytope  $P$  it is a symmetry for. Further,  $S_n^\neg := S_n \circ S^\neg = S^\neg \circ S_n$  is a group by Lemma 3.7.

Consider the so-called *hypercube*  $Q_n \subset \mathbb{R}^n$  given by its  $\mathcal{V}$ -representation

$$Q_n = \text{conv}(\{0, 1\}^n),$$

where  $\text{vert}(Q_n) = \{0, 1\}^n$ . The facets of  $Q_n$  are defined by the inequalities  $-x_i \leq 0$  and  $x_i \leq 1$  for  $i = 1, \dots, n$ . We get the following different equivalence classes with respect to the different symmetry groups  $S_n, S^\neg$  and  $S_n^\neg$ .

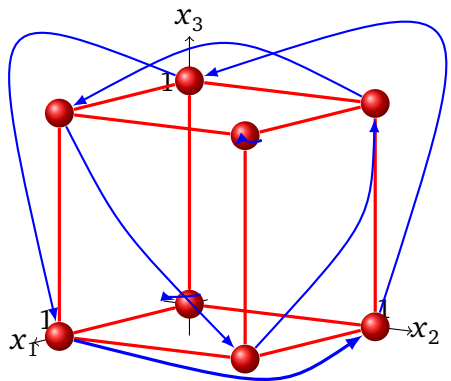


Figure 1: This figure shows where the symmetry  $(x_1, x_2, x_3) \mapsto (x_2, x_3, x_1)$  maps the vertices of the cube.

$S_n$  : As we can swap any two variables  $x_i$  and  $x_j$  with each other but cannot do anything about the form of the inequality, we get exactly two equivalence classes:

- $\{x \in P : x_i = 0\} : i \in \{1, \dots, n\}\}$ ,
- $\{x \in P : x_i = 1\} : i \in \{1, \dots, n\}\}$ .

See Figures 1–3 for a visualization.

$S_n^-$  : Using  $s_{\{i\}}^-$  one can transform the facet  $\{x \in P : -x_i = 0\}$  into the facet  $\{x \in P : x_i = 1\}$ , but we cannot change which variables occur in the inequality. So we get a total of  $n$  equivalence classes, one for each  $i = 1, \dots, n$  of the form:

$$\{x \in P : -x_i = 0\}, \{x \in P : x_i = 1\}$$

See Figure 4 for a visualization.

$S_n^-$  : As we can swap any two variables  $x_i, x_j$  as well as use  $s_{\{i\}}^-$  to transform the facet-defining inequalities into one another, we only have one equivalence class which includes every facet of  $Q_n$ . Given the symmetrical nature of  $Q_n$ , this is exactly what we would like to see if we are interested in geometric properties.

This example shows that the equivalence classes depend on the chosen symmetry group.

## 4 Equivalence of Equations

Many combinatorial polytopes require equations for their linear description. However, we unfortunately cannot proceed in the same way as for inequalities to classify them. The reason is that the vertices are not an invariant of equations since every point in  $P$  has to satisfy them.

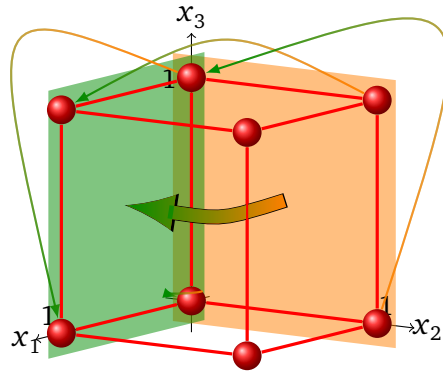


Figure 2: The symmetry  $(x_1, x_2, x_3) \mapsto (x_2, x_3, x_1)$  maps one facet of the cube onto another facet.

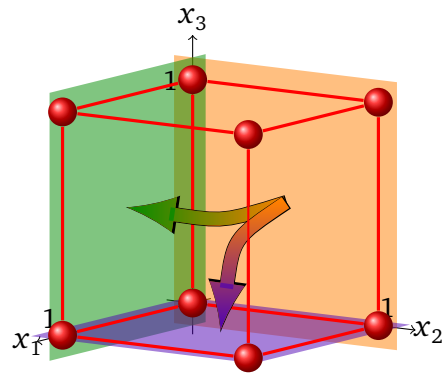


Figure 3: Shows where permutation symmetries could map a particular facet of the cube.

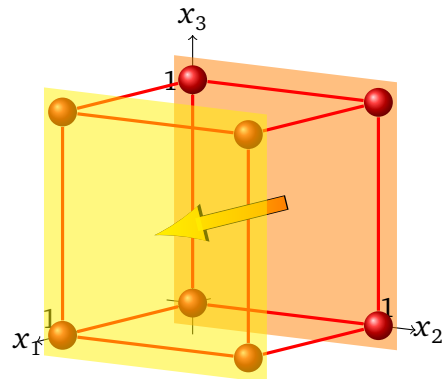


Figure 4: The symmetry  $(x_1, x_2, x_3) \mapsto (1 - x_1, x_2, x_3)$  maps one facet of the cube onto another facet. It cannot be mapped to this facet by a permutation symmetry.

After an example of what classification of equations means for the Asymmetric Traveling Salesman Problem, we will present a theoretical analysis of different invariants related to equations and group of symmetries.

**Example 4.1.** In Example 3.1, we analyzed the inequalities in the  $\mathcal{H}$ -representation of the Asymmetric Traveling Salesman Polytope. However, there are also  $n$  Equations (TSP i) and  $n$  Equations (TSP ii) present in the description of the polytope.

With the same symmetries as in Example 3.1, node permutations  $s^\sigma$  and the arc reversal  $r$ , these  $2n$  equations belong to one equivalence class: Using the node permutation  $\sigma$  that changes the nodes  $i_1 \in V_n$  and  $i_2 \in V_n$ , any two Equations (TSP i) with  $i = i_1$  and  $i = i_2$  belong to the same class. In the representation given, we can see this just by mapping the variables using the permutation symmetry  $\sigma$ . The same holds for the Equations (TSP ii) by interchanging the vertices  $j = j_1$  and  $j = j_2$ . Further, arc reversals imply that Equations (TSP i) and (TSP ii) for the same node  $i = j$  belong to the same class. Thus, the  $n$  Equations (TSP i) and  $n$  Equations (TSP ii) all belong to the same equivalence class.

However, there are many other equivalent representations of the Equations (TSP i) and (TSP ii). E. g., the given equations could be iteratively changed by adding a multiple of some equation to another one. Although this would of course not change the polytope, it would be not as easy as above to see that the equations belong to the same equivalence class.

We will now study different approaches to classify equations that might be given in different representations. Our results show that the simpler approaches do not lead to valid equivalence relations.

As a starting point, the next definition introduces the concept of a linear subspace that represents all the equations that can be generated by applying all symmetries from a given group to one equation. We will need this notion in our exploration of possible invariants for the identification of equations regardless of their representation.

**Definition 4.2.** Let  $S$  be a group of symmetries that are bijections on  $\mathbb{R}^n$  and  $c^T x = d$  a valid equation for the polytope  $P \subseteq \mathbb{R}^n$ . We can identify the equation with the vector  $(c, d) \in \mathbb{R}^{n+1}$ . Then the *symmetric subspace*  $U_S(c, d) \subset \mathbb{R}^{n+1}$  of  $(c, d)$  over  $S$  is defined as

$$U_S(c, d) := \text{span}(\{(c_\sigma, d_\sigma) : \sigma \in S, \sigma(x) = M_\sigma x + r_\sigma\}),$$

where  $c_\sigma := M^T c$  and  $d_\sigma = d - c^T r$ .

The following Lemma establishes an equivalence relation on equations.

**Lemma 4.3.** *The relation*

$$(c, d) \simeq_1 (c', d') : \Leftrightarrow U_S(c, d) = U_S(c', d')$$

*is an equivalence relation on the valid equations of a polytope  $P$ .*

*Proof.* Being an equivalence relation is a direct consequence of "=" being one:

- $(c, d) \simeq_1 (c, d)$  since  $U_S(c, d) = U_S(c, d)$ ,

- $(c, d) \simeq_1 (c', d') \Rightarrow (c', d') \simeq_1 (c, d)$   
since  $U_S(c, d) = U_S(c', d') \Rightarrow U_S(c', d') = U_S(c, d)$ , and
- $(c, d) \simeq_1 (c', d'), (c', d') \simeq_1 (c'', d'') \Rightarrow (c, d) \simeq_1 (c', d')$   
since  $U_S(c, d) = U_S(c', d'), U_S(c', d') = U_S(c'', d'') \Rightarrow U_S(c, d) = U_S(c'', d'')$ .

□

Lemma 4.3 shows an equivalence relation on equation. But  $\simeq_1$  does not necessarily produce a minimal number of equivalence-classes as we will show now. Consider the following example.

$$\begin{aligned}
S &:= \langle (2, 3), (4, 5) \rangle \subseteq S_5, \\
(c, d) &:= (1, 1, -1, 0, 0, 0), \\
(c', d') &:= (1, 0, 0, 0, 0, 0), \\
(c'', d'') &:= (1, 0, 0, 1, -1, 0).
\end{aligned}$$

Using the equivalence relation  $\simeq_1$ , we get three classes because we have  $(c, d) \not\simeq_1 (c', d') \not\simeq_1 (c'', d'') \not\simeq_1 (c, d)$ .

However, we might want to view  $(c, d)$ ,  $(c', d')$  and  $(c'', d'')$  as belonging to one equivalence class w. r. t.  $S$ . For each pair of these vectors  $(c^1, d^1), (c^2, d^2)$ , one of them lies in the symmetric subspace of the other one, i. e.,  $(c^1, d^1) \in U_S(c^2, d^2)$  or  $(c^2, d^2) \in U_S(c^1, d^1)$  holds.

Therefore, the result in the last Lemma is not satisfying. But simpler approaches do not lead to an equivalence relation, as we will show in the remainder of this section.

But at least we can use this approach to obtain some subclasses where all members are equivalent. As mentioned above, these classes are not necessarily complete.

For practical computations, we consider this result as not strong enough to be worth implementing. However, this is not such a big problem since most polytopes do not contain that many equations and their number is bounded by the number of variables.

The following lemma leads to further ideas for equivalence relations on equations.

**Lemma 4.4.** *The following implication is true:*

$$(c, d) \in U_S(c', d') \Rightarrow U_S(c, d) \subseteq U_S(c', d').$$

*Proof.*  $(c, d) \in U_S(c', d')$  means there is a representation

$$(c, d) = \sum_{\sigma \in S} \lambda_{\sigma} (c'_{\sigma}, d'_{\sigma})$$

where  $\lambda_{\sigma} \in \mathbb{R}$ . Now consider for  $\tau \in S$  the equation  $(c_{\tau}, d_{\tau}) \in U_S(c, d)$ . Using  $\tau$  on the

representation of  $(c, d)$  in  $U_S(c', d')$  yields

$$\begin{aligned}
(c_\tau, d_\tau) &= \sum_{\sigma \in S} \lambda_\sigma((c'_\sigma)_\tau, (d'_\sigma)_\tau) \\
&= \sum_{\sigma \in S} \lambda_{(\tau^{-1} \circ \tau \circ \sigma)}(c'_{\tau \circ \sigma}, d'_{\tau \circ \sigma}) \\
&= \sum_{\sigma \in \tau(S)} \lambda_{(\tau^{-1} \circ \sigma)}(c'_\sigma, d'_\sigma) \\
&= \sum_{\sigma \in S} \lambda_{(\tau^{-1} \circ \sigma)}(c'_\sigma, d'_\sigma) \in U_S(c', d').
\end{aligned}$$

□

Unfortunately, the implication in Lemma 4.4 does not result in an equivalence relation, as we show now.

**Lemma 4.5.** *The relation*

$$(c, d) \simeq_2 (c', d') : \Leftrightarrow (c, d) \in U_S(c', d')$$

*is not an equivalence relation on the valid equations of a polytope  $P$ .*

*Proof.* Although the relation is reflexive and transitive, it may not be symmetric; a counterexample is easy to construct:

$$S := \langle (2, 3) \rangle \subseteq S_3, \quad (c, d) := (1, 0, 0, 0), \quad (c', d') := (1, 1, -1, 0).$$

□

To avoid the problem that the relation is not symmetric, we try a weaker approach.

**Lemma 4.6.** *The relation*

$$(c, d) \simeq_3 (c', d') : \Leftrightarrow (c, d) \in U_S(c', d') \text{ or } (c', d') \in U_S(c, d)$$

*is not an equivalence relation on the valid equations of a polytope  $P \subset \mathbb{R}^n$ .*

*Proof.* Although the relation is reflexive and symmetric, it may not be transitive. A counterexample where  $(c, d) \simeq_3 (c', d')$ ,  $(c', d') \simeq_3 (c'', d'')$  but  $(c, d) \not\simeq_3 (c'', d'')$  is again the one we used in the discussion of the result in Lemma 4.3,

$$\begin{aligned}
S &:= \langle (2, 3), (4, 5) \rangle \subseteq S_5, \\
(c, d) &:= (1, 1, -1, 0, 0, 0), \\
(c', d') &:= (1, 0, 0, 0, 0, 0), \\
(c'', d'') &:= (1, 0, 0, 1, -1, 0).
\end{aligned}$$

□

So this approach also does not work. The only possibility seems to be to define an equivalence relation based on Definition 4.2 and to demand the equivalence of the two spans themselves, as shown in Lemma 4.3.



## 5 A Facet Classification Algorithm

Based on the results of the previous sections we propose the following algorithm for facet classification.

---

### Algorithm 1: HUHFA

---

**Data:** *min.*  $\mathcal{H}$ ,  $\mathcal{V}$ -representations of polytope  $P \subset \mathbb{R}^n$  and symmetries given by

- list of vertices  $v_1, v_2, \dots, v_k$ ,
- list of facet-defining inequalities  $f_1, f_2, \dots, f_m$ ,
- set  $S = \{s_1, s_2, \dots, s_l\}$  of symmetries on  $P$ ,
- Boolean closed indicating whether  $(S, \circ)$  is a group.

**Result:** equivalence classes Class of facets of  $P$  w. r. t.  $\langle S \rangle$

```

1 Incidence  $\leftarrow$  two-dimensional array of size  $m \times k$ 
2 P  $\leftarrow$  two-dimensional array of size  $l \times k$ 
3 Class  $\leftarrow$  array of size  $m$ 
4 Index  $\leftarrow$  empty list of key-value pairs (array with  $k$  entries (incidence vector of facet),
  integer) lexicographically sorted by the keys
5 VertIndex  $\leftarrow$  list of key-value pairs (vertex  $v_i$ ,  $i$ ) lexicographically sorted by the keys
6 PermIndex  $\leftarrow$  array of size  $k$ 

7 for  $i \leftarrow 1$  to  $m$  do // For all Facet-defining inequalities
8   for  $o \leftarrow 1$  to  $k$  do // For all vertices
9     Incidence[ $i, o$ ]  $\leftarrow$  1 if  $v_o$  satisfies the inequality  $f_i$  with equality, 0 otherwise
10    // create incidence vector
11  Index( Incidence[ $i$ ]) $\leftarrow$  $i$  // index the vector to identify it quickly
12  Class[ $i$ ] $\leftarrow$  $i$  // set class of facet  $f_i$  to  $i$ 
13 for  $j \leftarrow 1$  to  $l$  do // For all symmetries
14   for  $o \leftarrow 1$  to  $k$  do // For all vertices
15    P[ $j, o$ ] $\leftarrow$  VertIndex( $s_j(v_o)$ ) // create mapping table
16 for  $i \leftarrow 1$  to  $m$  do // for each facet-defining inequality  $f_1, f_2, \dots, f_m$ 
17   if not closed or Class[ $i$ ] $\neq$  $i$  then
18     for  $j \leftarrow 1$  to  $l$  do // for each symmetry  $s_1, s_2, \dots, s_l$ 
19       for  $o \leftarrow 1$  to  $k$  do // for each vertex  $v_1, v_2, \dots, v_k$ 
20        PermIndex[ $o$ ]=Incidence[ $i, P[j, o]$ ] // create vector by using  $j$ 
21        // on the incidence vector of  $j$ 
22        ImageIndex  $\leftarrow$  Index(PermIndex) // identify class
23        unite classes  $i$  and ImageIndex in Class // with a disjoint-set data
24        structure maintained and updated in parallel to Class
25 return Class

```

---

**Theorem 5.1.** *Algorithm 1 works correctly and terminates.*

*Proof.* We first prove that the algorithm is well-defined. It is obvious that it then also terminates.

The only critical part is Step 20 where it is not immediately clear that  $\text{PerIndex}$  is the incidence vector of a facet of  $P$ . However, because the  $s_j$  is a symmetry for  $P$ , this property follows directly from Theorem 2.6.

Thus, the proof of termination is completed and we proceed with the correctness. We will need the following definition. Let  $G_{P,S} = (V, E)$  denote the graph of facet symmetries of  $P$  with regard to  $S$ , i. e.,  $V = \{1, \dots, m\}$  where  $i \in V$  refers to the facet  $F_i$  induced by  $f_i$ , and  $\{i, j\} \in E$  if and only if there exists a map  $s \in S$  such that either  $f_i$  is equivalent to  $f_j$  with regard to  $s$  or vice versa. Note that we allow loops to arise.

To show correctness, we first prove that the algorithm computes the connected components of  $G_{P,S}$  such that the corresponding facet-defining inequalities from one component all have the same Class in the algorithm.

Suppose that  $S$  is closed. If we have both edges  $\{a, b\}$  and  $\{b, c\}$  in  $E$  it follows from closure that we have the edge  $\{a, c\}$  as well. Using this argument iteratively on every path connecting nodes  $i$  and  $j$  shows that already  $\{i, j\} \in E$  which implies that the connected components form complete subgraphs. In this case, we get the whole connected component of some  $i \in V$  by examining all its incident edges, which is exactly what the algorithm does in Steps 15–21.

So let  $S$  be not closed. This means that we have to check the neighborhood of every node in order to compute the connected components, which again is exactly what the algorithm does in Steps 15–21.

Therefore, the main argument for correctness in the case in which  $S$  is not closed lies in the following claim:  $G_{P,S}$  and  $G_{P,\langle S \rangle}$  have the same connected components.

We may view the maps in  $S$  as permutations of the set  $\text{vert}(P)$  where their explicit form as affine maps is only used to encode these vertex permutations. (In fact, this is what the algorithm does in Steps 12–14. However, saving the maps as vertex permutations is not necessary but is supposed to reduce the running time.) Then we can view  $\langle S \rangle$  as a finite group. Finiteness is justified because we retain all the information needed to determine the edge sets.

Obviously, because  $S \subseteq \langle S \rangle$ , every pair of nodes connected in  $G_{P,S}$  is connected in  $G_{P,\langle S \rangle}$  as well. So suppose that  $i \in V$  and  $j \in V$  are connected in  $G_{P,\langle S \rangle}$ . As stated previously, the connected components of  $G_{P,\langle S \rangle}$  form complete subgraphs such that  $\{i, j\}$  is an edge in  $G_{P,\langle S \rangle}$ . By our definition, this means that there exists a map  $s_{ij} \in \langle S \rangle$  such that  $F_i$  and  $F_j$  are equivalent with respect to  $s_{ij}$ .

By definition of  $\langle S \rangle$  this means that  $s_{ij}$  can be written as the composition of a sequence of maps from  $S$ . Because  $\langle S \rangle$  is finite, we can assume this sequence to be finite as well. Suppose that  $s_{ij} = \sigma_t \circ \dots \circ \sigma_2 \circ \sigma_1$  where  $\sigma_r \in S$  for  $r = 1, \dots, t$ . By Theorem 2.6, we have that  $F^{(r)} := (\sigma_r \circ \dots \circ \sigma_2 \circ \sigma_1)(F_i)$  is a facet of  $P$  as well. In this setting  $F^{(t)} = F_j$ ,  $F^{(0)} = F_i$  and we have that  $F^{(r+1)} = \sigma_{r+1}(F^{(r)})$  for  $r = 0, \dots, t - 1$ . But now it follows from our definition of the edge set that this does in fact define a path connecting  $i$  and  $j$  in  $G_{P,S}$ . Therefore the

claim and the theorem are proven. □

**Theorem 5.2.** *Algorithm 1 has a worst case running time of*

$$O(m(nk + \log(m)) + kl(n^2 + \log(k)) + ml(k + \log(m)))$$

*if  $S$  is not closed, and*

$$O(m(nk + \log(m)) + kl(n^2 + \log(k)) + cl(k + \log(m)))$$

*if  $S$  is closed, where  $c$  denotes the number of equivalence classes with respect to  $S$ .*

*Proof.* Running time of Steps 1–6 can be neglected.

Loop 7 has  $m$  iterations, Loop 8 has  $k$  iterations. Step 9 consists of at most  $n$  multiplications and  $n - 1$  addition operations. Step 10 adds an element to a sorted list with  $\leq m$  entries, this is in  $O(\log(m))$ . Step 11 is in  $O(1)$ . Thus, Steps 7–11 are in  $O(m(kn + \log(m)))$ .

Loops 12 and 13 have  $l$  and  $k$  iterations, respectively. Step 14 involves the evaluation of an affine map, which can be done in  $O(n^2)$ , and finding an element in a sorted list with  $k$  entries, which can be done in  $O(\log(k))$ . All other operations in this step are in  $O(1)$ . Thus, Steps 12–14 have a running time in  $O(kl(n^2 + \log(k)))$ .

There are  $m$  iterations of Loop 15,  $l$  iterations of Loop 17 and  $k$  iterations of Loop 18. Step 19 is in  $O(1)$  and Step 20 in  $O(\log(m))$ . Because of the use of a disjoint-set data structure, Step 21 has an amortized constant of  $\alpha(m)$ , where  $\alpha(m)$  denotes the inverse of the Ackermann function (this value does not rise above 4 for all practical purposes, [Cormen et al., 1989]).

If  $S$  is closed we enter Step 21 exactly once for every class and we get the estimate of  $O(cl(k + \log(m) + \alpha(m))) = O(cl(k + \log(m)))$  for Steps 15–21 where  $c$  denotes the number of equivalence classes with respect to  $S$ . Otherwise we do Step 21 once for every facet and get  $O(ml(k + \log(m) + \alpha(m))) = O(ml(k + \log(m)))$  for these steps.

All together, these terms sum up to

$$O(m(nk + \log(m)) + kl(n^2 + \log(k)) + al(k + \log(m)))$$

where  $a = c$  if  $S$  is closed and  $a = m$  else. □

However, with reasonable assumptions the term can be simplified:

**Corollary 5.3.** *Assuming  $\log(m) \leq k \leq m$ ,  $n \leq k$ ,  $n \in O(m)$ , sparsity of the affine maps in  $S$ , that is, they can be applied to the vertices in linear time, and  $|S| \in O(k)$ , Algorithm 1 has a running time in  $O(mk^2)$ .*

These assumptions are reasonable in the sense that they are often encountered when dealing with combinatorial problems and their LP-formulations. In this scenario, symmetries are often given by permuting variables or flipping their respective 0/1-values, which

Polytope	#Variables	#Vertices	#Facets	$ S  /   \langle S \rangle  $	closed	#Classes	CPU (in s)
$P_{LO}^6$	15	720	910	6 / ??	false	2	2.2
$P_{TSP}^5$	20	24	390	5 / 240	false	6	0.8
$P_{TVP}^4$	18	24	1280	4 / 48	false	48	0.2
$P_{STVP}^5$	30	120	30040	5 / 240	false	175	17.2
$P_{HAP}^3$	45	978	14049	4608 / 4608	true	30	90.3

Table 1: Run times of HUHFA for different combinatorial optimization problem polytopes.

naturally results in sparse matrices. At the same time, it is a well known fact that every subgroup of the symmetric group  $S_k$  can be generated by at most  $k$  maps, and because  $\langle S \rangle$  is a subgroup of  $S_k$  (it permutes  $k$  vertices), this applies to  $\langle S \rangle$  as well. Thus generators can be chosen accordingly, and there are known algorithms for this task which run in polynomial time [Jerrum, 1986]. The difficulty would be to find a low number of generators which possess a sparse affine map presentation at the same time but as said before, this case arises naturally in practice. The estimates for  $m$  and  $k$  and their relation to  $n$  are typical for NP-hard problems.

With an implementation of Algorithm 1 we were able to produce the results shown in Table 1. The algorithm was implemented in C++ and all computations were performed on an Intel(R) Core(TM) i7-2600 CPU 4-core processor with 3.40 GHz and 16 GB Ram.

With our computations, we contributed to the understanding of the polyhedral structure of certain combinatorial optimization problems with a large number of facets. Although we could calculate all the facet-defining inequalities from the known  $\mathcal{V}$ -descriptions before, classifying the inequalities manually was not possible. Further, we could show that Algorithm 1 results in a practically usable software.

As test instances we used several LOP and TSP polytopes as well as polytopes describing the symmetric and asymmetric Target Visitation Problem (STVP/TVP) [Hildenbrandt and Reinelt, ], a combination of the Linear Ordering Problem and the Traveling Salesman Problem, or the Hypergraph Assignment Problem (HAP) for part size two [Borndörfer and Heismann, 2012], a generalization of the assignment problem to bipartite hypergraphs. Both TVP, STVP and HAP are problems where already very small instances are described by a very large number of facets.

Table 1 shows the computation times of our algorithm for these problems. The chosen problem sizes are the maximum ones which PORTA is able to compute in a reasonable amount of time. As you can see, HUHFA can classify the facets for the different problems that we have included in our computations in a practically acceptable time.

The results obtained with HUHFA lead us a better understanding of the polytopes for the Target Visitation and Hypergraph Assignment Problems. For example, we realized that one half of the 30 facet classes for  $P_{HAP}^3$  can be described combinatorially as a generalization

of odd set inequalities for the matching problem. This would have not been possible if we had to deal with the more than 14,000 facets. Using HUHFA, we could also easily check that all the 14049 facet-defining inequalities can be stated with coefficients -1, 0 and 1, and a right hand side equal to 1. We do not know whether this result holds in general for the Hypergraph Assignment Problem, however, it is an interesting starting point for further research in this direction.

The software HUHFA can be downloaded from the website <http://comopt.ifi.uni-heidelberg.de/people/hildenbrandt/HUHFA>.

## 6 Conclusion

In this paper, we presented new ideas for classifying facets of a given polytope with respect to given symmetries. Contrary to all approaches known so far, this method does not need any normal forms and can be applied to all kinds of combinatorial polytopes. Our theoretical results show that the vertices which are met by facets with equality can be, in fact, used as invariants. Our computations suggest that our method can be used without time problems in practice for all polytopes which PORTA is able to compute. Further, the results of our computations allowed important insights into the facet structure of certain polytopes.

An open problem that remains is to find a practically applicable way to classify equalities and implement it.

## References

- [Bolotashvili et al., 1999] Bolotashvili, G., Kovalev, M., and Girlich, E. (1999). New facets of the linear ordering polytope. *SIAM Journal on Discrete Mathematics*, 12(3):326–336.
- [Borndörfer and Heismann, 2012] Borndörfer, R. and Heismann, O. (2012). Minimum cost hyperassignments with applications to ICE/IC rotation planning. In *Operations Research Proceedings 2011*, pages 59–64. Springer Verlag.
- [Christof and Loebel, 2008] Christof, T. and Loebel, A. (2008). PORTA: Polyhedron representation transformation algorithm, version 1.4. <http://comopt.ifi.uni-heidelberg.de/software/PORTA/index.html>.
- [Christof and Reinelt, 2001] Christof, T. and Reinelt, G. (2001). Decomposition and parallelization techniques for enumerating the facets of combinatorial polytopes. *Int. J. Comput. Geometry Appl.*, 11(4):423–437.
- [Cormen et al., 1989] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (1989). *Introduction to Algorithms*. The MIT Press and McGraw-Hill Book Company.
- [Hildenbrandt and Reinelt, ] Hildenbrandt, A. and Reinelt, G. On the structure of the TVP polytope. In prep.

- [Jerrum, 1986] Jerrum, M. (1986). A compact representation for permutation groups. *Journal of Algorithms*, 7(1):60–78.
- [Naddef and Rinaldi, 1993] Naddef, D. and Rinaldi, G. (1993). The graphical relaxation: A new framework for the symmetric traveling salesman polytope. *Mathematical Programming*, 58(1-3):53–88.
- [Rehn, 2010] Rehn, T. (2010). Fundamental permutation group algorithms for symmetry computation. Diploma thesis, Otto von Guericke University Magdeburg. <http://fma2.math.uni-magdeburg.de/~latgeo/permlib/diploma-thesis-cs-rehn.pdf>.
- [Rehn and Schürmann, 2010] Rehn, T. and Schürmann, A. (2010). C++ tools for exploiting polyhedral symmetries. In Fukuda, K., Hoeven, J., Joswig, M., and Takayama, N., editors, *Mathematical Software – ICMS 2010*, volume 6327 of *Lecture Notes in Computer Science*, pages 295–298. Springer Berlin Heidelberg.
- [Reinelt, 1985] Reinelt, G. (1985). *The linear ordering problem: algorithms and applications*, volume 8. Heldermann Berlin.
- [Weyl, 1934] Weyl, H. (1934). Elementare Theorie der konvexen Polyeder. *Commentarii Mathematici Helvetici*, 7(1):290–306.