

# Some Algorithmic Problems in Polytope Theory

Volker Kaibel\* and Marc E. Pfetsch

TU Berlin  
MA 6-2  
Straße des 17. Juni 136  
10623 Berlin  
Germany  
{kaibel,pfetsch}@math.tu-berlin.de

## 1 Introduction

Convex polyhedra, i.e., the intersections of finitely many closed affine half-spaces in  $\mathbb{R}^d$ , are important objects in various areas of mathematics and other disciplines. In particular, the compact ones among them (*polytopes*), which equivalently can be defined as the convex hulls of finitely many points in  $\mathbb{R}^d$ , have been studied since ancient times (e.g., the platonic solids). Polytopes appear as building blocks of more complicated structures, e.g., in (combinatorial) topology, numerical mathematics, or computer aided design. Even in physics polytopes are relevant (e.g., in crystallography or string theory).

Probably the most important reason for the tremendous growth of interest in the theory of convex polyhedra in the second half of the 20<sup>th</sup> century was the fact that *linear programming* (i.e., optimizing a linear function over the solutions of a system of linear inequalities) became a widespread tool to solve practical problems in industry (and military). Dantzig's Simplex Algorithm, developed in the late 40's, showed that geometric and combinatorial knowledge of polyhedra (as the domains of linear programming problems) is quite helpful for finding and analyzing solution procedures for linear programming problems.

Since the interest in the theory of convex polyhedra to a large extent comes from algorithmic problems, it is not surprising that many algorithmic questions on polyhedra arose in the past. But also inherently, convex polyhedra (in particular: polytopes) give rise to algorithmic questions, because they can be treated as finite objects by definition. This makes it possible to investigate (the smaller ones among) them by computer programs (like the `polymake`-system written by Gawrilow and Joswig, see [26] and [27,28]). Once chosen to exploit this possibility, one immediately finds oneself confronted with many algorithmic challenges.

This paper contains descriptions of 35 algorithmic problems about polyhedra. The goal is to collect for each problem the current knowledge about its

---

\* Supported by the Deutsche Forschungsgemeinschaft, FOR 413/1-1 (Zi 475/3-1).

computational complexity. Consequently, our treatment is focused on theoretical rather than on practical subjects. We would, however, like to mention that for many of the problems computer codes are available.

Our choice of problems to be included is definitely influenced by personal interest. We have not spent particular efforts to demonstrate for each problem why we consider it to be relevant. It may well be that the reader finds other problems at least as interesting as the ones we discuss. We would be very interested to learn about such problems. The collection of problem descriptions presented in this paper is intended to be maintained as a (hopefully growing) list at <http://www.math.tu-berlin.de/~pfetsch/polycomplex/>.

Almost all of the problems are questions about polytopes. In some cases the corresponding questions on general polyhedra are interesting as well. It can be tested in polynomial time whether a polyhedron specified by linear inequalities is bounded or not. This can be done by applying Gaussian elimination and solving one linear program.

Roughly, the problems can be divided into two types: problems for which the input are “geometrical” data and problems for which the input is “combinatorial” (see below). Actually, it turned out that it was rather convenient to group the problems we have selected into the five categories “Coordinate Descriptions” (Sect. 2), “Combinatorial Structure” (Sect. 3), “Isomorphism” (Sect. 4), “Optimization” (Sect. 5), and “Realizability” (Sect. 6). Since the boundary complex of a simplicial polytope is a simplicial complex, studying polytopes leads to questions that are concerned with more general (polyhedral) structures: *simplicial complexes*. Therefore, we have added a category “Beyond Polytopes” (Sect. 7), where a few problems concerned with general (abstract) simplicial complexes are collected that are closely related to similar problems on polytopes. We do not consider other related areas like oriented matroids.

The problem descriptions proceed along the following scheme. First input and output are specified. Then a summary of the knowledge on the theoretical complexity is given, e.g., it is stated that the complexity is unknown (“Open”) or that the problem is  $\mathcal{NP}$ -hard. This is done for the case where the dimension (usually of the input polytope) is part of the input as well as for the case of fixed dimension; often the (knowledge on the) complexity status differs for the two versions. After that, comments on the problems are given together with references. For each problem we tried to report on the current state of knowledge according to the literature. Unless stated otherwise, all results mentioned without citations are either considered to be “folklore” or “easy to prove.” At the end related problems in this paper are listed.

For all notions in the theory of polytopes that we use without explanation we refer to Ziegler’s book [65]. Similarly, for the concepts from the theory of computational complexity that play a role here we refer to Garey and Johnson’s classical text [24]. Whenever we talk about *polynomial reductions* this refers to polynomial time Turing-reductions. For some of the problems

the output can be exponentially large in the input. For these problems the interesting question is whether there is a *polynomial total time* algorithm, i.e., an algorithm whose running time can be bounded by a polynomial in the sizes of the input and the output (in contrast to a *polynomial time* algorithm whose running time would be bounded by a polynomial just in the input size). Note that the notion of “polynomial total time” only makes sense with respect to problems which explicitly require the output to be non-redundant.

A very fundamental result in the theory of convex polyhedra is due to Minkowski [46] and Weyl [64]. For the special case of polytopes (to which we restrict our attention from now on) it can be formulated as follows. Every polytope  $P \subset \mathbb{R}^d$  can be specified by an  $\mathcal{H}$ - or by a  $\mathcal{V}$ -description. Here, an  $\mathcal{H}$ -description consists of a finite set of linear inequalities (defining closed affine half-spaces of  $\mathbb{R}^d$ ) such that  $P$  is the set of all simultaneous solutions to these inequalities. A  $\mathcal{V}$ -description consists of a finite set of points in  $\mathbb{R}^d$  whose convex hull is  $P$ . If any of the two descriptions is rational, then the other one can be chosen to be rational as well. Furthermore, in this case the numbers in the second description can be chosen such that their coding lengths depend only polynomially on the coding lengths of the numbers in the first description (see, e.g., Schrijver [55]). In our context,  $\mathcal{H}$ - and  $\mathcal{V}$ -descriptions are usually meant to be rational. By linear programming, each type of description can be made non-redundant in polynomial time (though it is unknown whether this is possible in *strongly* polynomial time, see Problem 24).

One of the basic properties of a polytope is its dimension. If the polytope is given by a  $\mathcal{V}$ -description, then it can easily be determined by Gaussian elimination (which, carefully done, is a cubic algorithm; see, e.g., [55]). If the polyhedron is specified by an  $\mathcal{H}$ -description, computing its dimension can be done by linear programming (actually, this is polynomial time equivalent to linear programming).

Furthermore, some of the problems may also be interesting in their polar formulations, i.e., with “the roles of  $\mathcal{H}$ - and  $\mathcal{V}$ -descriptions exchanged.” Switching to the polar requires to have a relative interior point at hand, which is easy to obtain if a  $\mathcal{V}$ -description is available, while it needs linear programming if only an  $\mathcal{H}$ -description is specified.

We will especially be concerned with the combinatorial types of polytopes, i.e., with their *face lattices* (the sets of faces, ordered by inclusion). In particular, some problems will deal with the *k-skeleton* of a polytope, which is the set of its faces of dimensions less than or equal to  $k$ , or with its *f-vector*, i.e., the vector  $(f_0(P), f_1(P), \dots, f_d(P))$ , where  $f_i(P)$  is the number of  $i$ -dimensional faces (*i-faces*) of the  $d$ -dimensional polytope  $P$  (*d-polytope*). Talking of the face lattice  $\mathcal{L}_P$  of a polytope  $P$  will always refer to the lattice as an abstract object, i.e., to *any* lattice that is isomorphic to the face lattice. In particular, the lattice does not contain any information on coordinates. Similarly, the *vertex-facet incidences* of  $P$  are given by any matrix  $(a_{vf})$  with entries from  $\{0, 1\}$ , whose rows and columns are indexed by the vertices and facets

of  $P$ , respectively, such that  $a_{vf} = 1$  if and only if vertex  $v$  is contained in facet  $f$ . Note that the vertex-facet incidences of a polytope completely determine its face lattice.

A third important combinatorial structure associated with a polytope  $P$  is its (abstract) graph  $\mathcal{G}_P$ , i.e., any graph that is isomorphic to the graph having the vertices of  $P$  as its nodes, where two of them are adjacent if and only if their convex hull is a (one-dimensional) face of  $P$ . For simple polytopes, the (abstract) graph determines the entire face lattice as well (see Problem 15). However, for general polytopes this is not true.

Throughout the paper,  $n$  refers to the number of vertices or points in the given  $\mathcal{V}$ -description, respectively, depending on the context. Moreover,  $m$  refers to the number of facets or inequalities in the given  $\mathcal{H}$ -description, respectively, and  $d$  refers to the dimension of the polytope or the ambient space, respectively.

**Acknowledgment:** We thank the referee for many valuable comments and Günter M. Ziegler for carefully reading the manuscript.

## 2 Coordinate Descriptions

In this section problems are collected whose input are geometrical data, i.e., the  $\mathcal{H}$ - or  $\mathcal{V}$ -description of a polytope. Some problems which are also given by geometrical data appear in Sections 4 and 5.

### 1. VERTEX ENUMERATION

**Input:** Polytope  $P$  in  $\mathcal{H}$ -description

**Output:** Non-redundant  $\mathcal{V}$ -description of  $P$

**Status (general):** Open; polynomial total time if  $P$  is simple or simplicial

**Status (fixed dim.):** Polynomial time

Let  $d = \dim(P)$  and let  $m$  be the number of inequalities in the input. It is well known that the number of vertices  $n$  can be exponential ( $\Omega(m^{\lfloor d/2 \rfloor})$ ) in the size of the input (e.g., Cartesian products of suitably chosen two-dimensional polytopes and prisms over them).

VERTEX ENUMERATION is strongly polynomially equivalent to Problem 3 (see Avis, Bremner, and Seidel [1]). Since Problem 2 is strongly polynomially equivalent to Problem 3 as well, VERTEX ENUMERATION is also strongly polynomially equivalent to Problem 2.

For fixed  $d$ , Chazelle [12] found an  $\mathcal{O}(m^{\lfloor d/2 \rfloor})$  polynomial time algorithm, which is optimal by the Upper Bound Theorem of McMullen [43]. There exist algorithms which are faster than Chazelle's algorithm for small  $n$ , e.g., an  $\mathcal{O}(m \log n + (mn)^{1-1/(\lfloor d/2 \rfloor + 1)})$  polylog  $m$  algorithm of Chan [9].

For general  $d$ , the reverse search method of Avis and Fukuda [2] solves the problem for *simple* polyhedra in polynomial total time, using working space

(without space for output) bounded polynomially in the input size. An algorithm of Bremner, Fukuda, and Marzetta [8] solves the problem for *simplicial* polytopes. Note that these algorithms need a vertex of  $P$  to start from. Provan [52] gives a polynomial total time algorithm for enumerating the vertices of polyhedra arising from networks.

There are many more algorithms known for this problem – none of them is a polynomial total time algorithm for general polytopes. See the overview article of Seidel [57]. Most of these algorithms can be generalized to directly work for unbounded polyhedra, too.

**Related problems:** 2, 3, 5, 7

## 2. FACET ENUMERATION

**Input:** Polytope  $P$  in  $\mathcal{V}$ -description with  $n$  points

**Output:** Non-redundant  $\mathcal{H}$ -description of  $P$

**Status (general):** Open; polynomial total time if  $P$  is simple or simplicial

**Status (fixed dim.):** Polynomial time

In [1] it is shown that FACET ENUMERATION is strongly polynomially equivalent to Problem 3 and thus to Problem 1 (see the comments there).

For this problem, one can assume to have an interior point (e.g., the vertex barycenter). FACET ENUMERATION is sometimes called the *convex hull problem*.

**Related problems:** 1, 3, 5

## 3. POLYTOPE VERIFICATION

**Input:** Polytope  $P$  given in  $\mathcal{H}$ -description, polytope  $Q$  given in  $\mathcal{V}$ -description

**Output:** “Yes” if  $P = Q$ , “No” otherwise

**Status (general):** Open; polynomial time if  $P$  is simple or simplicial

**Status (fixed dim.):** Polynomial time

POLYTOPE VERIFICATION is strongly polynomially equivalent to Problem 1 and Problem 2 (see the comments there).

POLYTOPE VERIFICATION is contained in  $\text{co}\mathcal{NP}$ : we can prove  $Q \not\subseteq P$  by showing that some vertex of  $Q$  violates one of the inequalities describing  $P$ . If  $Q \subset P$  with  $Q \neq P$  then there exists a point  $p$  of  $P \setminus Q$  with “small” coordinates (e.g., some vertex of  $P$  not contained in  $Q$ ) and a valid inequality for  $Q$ , which has “small” coefficients and is violated by  $p$  (e.g., an inequality defining a facet of  $Q$  that separates  $p$  from  $Q$ ). However, it is unknown whether POLYTOPE VERIFICATION is in  $\mathcal{NP}$ .

Since it is easy to check whether  $Q \subseteq P$ , POLYTOPE VERIFICATION is Problem 4 restricted to the case that  $Q \subseteq P$ .

**Related problems:** 1, 2, 4

## 4. POLYTOPE CONTAINMENT

**Input:** Polytope  $P$  given in  $\mathcal{H}$ -description, polytope  $Q$  given in  $\mathcal{V}$ -description

**Output:** “Yes” if  $P \subseteq Q$ , “No” otherwise

**Status (general):**  $\text{co}\mathcal{NP}$ -complete

**Status (fixed dim.):** Polynomial time

Freund and Orlin [20] proved that this problem is  $\text{co}\mathcal{NP}$ -complete. Note that the reverse question whether  $Q \subseteq P$  is trivial. The questions where either both  $P$  and  $Q$  are given in  $\mathcal{H}$ -description or both in  $\mathcal{V}$ -description can be solved by linear programming (Problem 24), see Eaves and Freund [17]. For fixed dimension, one can enumerate all vertices of  $P$  in polynomial time (see Problem 1) and compare the descriptions of  $P$  and  $Q$  (after removing redundant points).

**Related problems:** 3

## 5. FACE LATTICE OF GEOMETRIC POLYTOPES

**Input:** Polytope  $P$  in  $\mathcal{H}$ -description

**Output:** Hasse-diagram of the face lattice of  $P$

**Status (general):** Polynomial total time

**Status (fixed dim.):** Polynomial time

See comments on Problem 1. Many algorithms for the VERTEX ENUMERATION PROBLEM in fact compute the whole face lattice of the polytope. Swart [60], analyzing an algorithm of Chand and Kapur [10], proved that there exists a polynomial total time algorithm for this problem. For a faster algorithm see Seidel [56]. Fukuda, Liebling, and Margot [22] gave an algorithm which uses working space (without space for the output) bounded polynomially in the input size, but it has to solve many linear programs. For fixed dimension, the size of the output is polynomial in the size of the input; hence, a polynomial total time algorithm becomes a polynomial algorithm in this case.

The problem of enumerating the  $k$ -skeleton of  $P$  seems to be open, even if  $k$  is fixed. Note that, for fixed  $k$ , the latter problem can be solved by linear programming (Problem 24) in polynomial time if the polytope is given in  $\mathcal{V}$ -description rather than in  $\mathcal{H}$ -description.

**Related problems:** 1, 2, 3, 13, 14

## 6. DEGENERACY TESTING

**Input:** Polytope  $P$  in  $\mathcal{H}$ -description

**Output:** “Yes” if  $P$  not simple (degenerate), “No” otherwise

**Status (general):** Strongly  $\mathcal{NP}$ -complete

**Status (fixed dim.):** Polynomial time

Independently proved to be  $\mathcal{NP}$ -complete in the papers of Chandrasekaran, Kabadi, and Murty [11] and Dyer [14]. Fukuda, Liebling, and Margot [22] proved that the problem is *strongly*  $\mathcal{NP}$ -complete. For fixed dimension, one can enumerate all vertices in polynomial time (see Problem 1) and check whether they are simple or not.

Bremner, Fukuda, and Marzetta [8] noted that if  $P$  is given in  $\mathcal{V}$ -description the problem is polynomial time solvable: enumerate the edges (1-skeleton, see Problem 5) and apply the Lower Bound Theorem.

Erickson [19] showed that in the worst case  $\Omega(m^{\lceil d/2 \rceil - 1} + m \log m)$  sidedness queries are required to test whether a polytope is simple. For odd  $d$  this matches the upper bound. A *sidedness query* is a question of the following kind: given  $d + 1$  points  $\mathbf{p}_0, \dots, \mathbf{p}_d$  in  $\mathbb{R}^d$ , does  $\mathbf{p}_0$  lie “above”, “below”, or on the oriented hyperplane determined by  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_d$ .

**Related problems:** 1, 5

## 7. NUMBER OF VERTICES

**Input:** Polytope  $P$  in  $\mathcal{H}$ -description

**Output:** Number of vertices of  $P$

**Status (general):**  $\#\mathcal{P}$ -complete

**Status (fixed dim.):** Polynomial time

Dyer [14] and Linial [40] independently proved that NUMBER OF VERTICES is  $\#\mathcal{P}$ -complete. It follows that the problem of computing the  $f$ -vector of  $P$  is  $\#\mathcal{P}$ -hard. Furthermore, Dyer [14] proved that the decision version (“Given a number  $k$ , does  $P$  have at least  $k$  vertices?”) is strongly  $\mathcal{NP}$ -hard and remains  $\mathcal{NP}$ -hard when restricted to simple polytopes. It is unknown whether the decision problem is in  $\mathcal{NP}$ .

If the dimension is fixed, one can enumerate all vertices in polynomial time (see Problem 1).

**Related problems:** 1, 14

## 8. FEASIBLE BASIS EXTENSION

**Input:** Polytope  $P$  given as  $\{\mathbf{x} \in \mathbb{R}^s : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ , a set  $S \subseteq \{1, \dots, s\}$

**Output:** “Yes” if there is a feasible basis with an index set containing  $S$ ,  
“No” otherwise

**Status (general):**  $\mathcal{NP}$ -complete

**Status (fixed dim.):** Polynomial time

See Murty [49] (Garey and Johnson [24], Problem [MP4]). For fixed dimension, one can enumerate all bases in polynomial time.

The problem can be reformulated as follows. Let  $P$  be defined by a finite set  $\mathcal{H}$  of affine halfspaces and let  $S$  be a subset of  $\mathcal{H}$ . Does  $\bigcap\{H \in \mathcal{H} : H \notin S\}$  contain a vertex which is also a vertex of  $P$ ?

## 9. RECOGNIZING INTEGER POLYHEDRA

**Input:** Polytope  $P$  in  $\mathcal{H}$ -description

**Output:** “Yes” if  $P$  has only integer vertices, “No” otherwise

**Status (general):** Strongly  $\text{co}\mathcal{NP}$ -complete

**Status (fixed dim.):** Polynomial time

The hardness-proof is by Papadimitriou and Yannakakis [51]. For fixed dimension, one can enumerate all vertices (Problem 1) and check whether they are integral in polynomial time.

## 10. DIAMETER

**Input:** Polytope  $P$  in  $\mathcal{H}$ -description

**Output:** The diameter of  $P$

**Status (general):**  $\mathcal{NP}$ -hard

**Status (fixed dim.):** Polynomial time

Frieze and Teng [21] gave the proof of  $\mathcal{NP}$ -hardness. For fixed dimension, one can enumerate all vertices (Problem 1), construct the graph and then compute the diameter in polynomial time.

The complexity status is unknown for simple polytopes. For simplicial polytopes the problem can be solved in polynomial time: Since simplicial polytopes have at most as many vertices as facets, one can enumerate their vertices (see Problem 1), and finally compute the graph (and hence the diameter) from the vertex-facet incidences in polynomial time.

If  $P$  is given in  $\mathcal{V}$ -description, one can compute the graph (1-skeleton, see Problem 5) and hence the diameter in polynomial time.

## 11. MINIMUM TRIANGULATION

**Input:** Polytope  $P$  in  $\mathcal{V}$ -description, positive integer  $K$

**Output:** “Yes” if  $P$  has a triangulation of size  $K$  or less, “No” otherwise

**Status (general):**  $\mathcal{NP}$ -complete

**Status (fixed dim.):**  $\mathcal{NP}$ -complete

A *triangulation*  $\mathcal{T}$  of a  $d$ -polytope  $P$  is a collection of  $d$ -simplices, whose union is  $P$ , their vertices are vertices of  $P$ , and any two simplices intersect in a common face (which might be empty). In particular,  $\mathcal{T}$  is a (pure)  $d$ -dimensional geometric simplicial complex (see Section 7). The size of  $\mathcal{T}$  is the number of its  $d$ -simplices. Every (convex) polytope admits a triangulation. Below, De Loera, and Richter-Gebert [4,5] proved that MINIMUM TRIANGULATION is  $\mathcal{NP}$ -complete for (fixed)  $d \geq 3$ . Furthermore, it is  $\mathcal{NP}$ -hard to compute a triangulation of minimal size for (fixed)  $d \geq 3$ .



## 12. VOLUME

**Input:** Polytope  $P$  in  $\mathcal{H}$ -description**Output:** The volume of  $P$ **Status (general):**  $\#\mathcal{P}$ -hard, FPRAS**Status (fixed dim.):** Polynomial time

Dyer and Frieze [15] showed that the general problem is  $\#\mathcal{P}$ -hard (and  $\#\mathcal{P}$ -easy as well). Dyer, Frieze, and Kannan [16] found a *Fully Polynomial Randomized Approximation Scheme (FPRAS)* for the problem, i.e., a family  $(A_\varepsilon)_{\varepsilon>0}$  of randomized algorithms, where, for each  $\varepsilon > 0$ ,  $A_\varepsilon$  computes a number  $V_\varepsilon$  with the property that the probability of  $(1 - \varepsilon) \text{vol}(P) \leq V_\varepsilon \leq (1 + \varepsilon) \text{vol}(P)$  is at least  $\frac{3}{4}$ , and the running times of the algorithms  $A_\varepsilon$  are bounded by a polynomial in the input size and  $\frac{1}{\varepsilon}$ .

For fixed dimension, one can first compute all vertices of  $P$  (see Problem 1) and its face lattice (see Problem 5) both in polynomial time. Then one can construct some triangulation (see Problem 11) of  $P$  (e.g., its barycentric subdivision) in polynomial time and compute the volume of  $P$  as the sum of the volumes of the (maximal) simplices in the triangulation.

The complexity status of the analogue problem with the polytope specified by a  $\mathcal{V}$ -description is the same.

## 3 Combinatorial Structure

In this section we collect problems that are concerned with computing certain combinatorial information from compact descriptions of the combinatorial structure of a polytope. Such compact encodings might be the vertex-facet incidences, or, for simple polytopes, the abstract graphs. An example of such a problem is to compute the dimension of a polytope from its vertex-facet incidences. Initialize a set  $S$  by the vertex set of an arbitrary facet. For each facet  $F$  compute the intersection of  $S$  with the vertex set of  $F$ . Replace  $S$  by a maximal one among the proper intersections and continue. The dimension is the number of “rounds” performed until  $S$  becomes empty.

All data is meant to be purely combinatorial. For all problems in this section it is unknown if the “integrity” of the input data can be checked, proved, or disproved in polynomial time. For instance, it is rather unlikely that one can efficiently prove or disprove that a lattice is the face lattice of some polytope (see Problems 29, 30).

Sometimes, it might be worthwhile to exchange the roles of vertices and facets by duality of polytopes. Our choices of view points have mainly been guided by personal taste.

Some orientations of the abstract graph  $\mathcal{G}_P$  of a simple polytope  $P$  play important roles (although such orientations can also be considered for non-simple polytopes, they have not yet proven to be useful in the more general

context). An orientation is called a *unique-sink orientation* (*US-orientation*) if it induces a unique sink on every subgraph of  $\mathcal{G}_P$  corresponding to a non-empty face of  $P$ . A US-orientation is called an *abstract objective function orientation* (*AOF-orientation*) if it is acyclic. General US-orientations of graphs of cubes have recently received some attention (Szabó and Welzl [61]). AOF-orientations were used, e.g., by Kalai [35]. Since their linear extensions are precisely the shelling orders of the dual polytope, they have been considered much earlier.

### 13. FACE LATTICE OF COMBINATORIAL POLYTOPES

**Input:** Vertex-facet incidence matrix of a polytope  $P$

**Output:** Hasse-diagram of the face lattice of  $P$

**Status (general):** Polynomial total time

**Status (fixed dim.):** Polynomial time

Solvable in  $\mathcal{O}(\min\{m, n\} \cdot \alpha \cdot \varphi)$  time, where  $m$  is the number of facets,  $n$  is the number of vertices,  $\alpha$  is the number of vertex-facet incidences, and  $\varphi$  is the size of the face lattice [33]. Note that  $\varphi$  is exponential in  $d$  (for fixed  $d$  it is polynomial in  $m$  and  $n$ ). Without (asymptotically) increasing the running time it is also possible to label each node in the Hasse diagram by the dimension and the vertex set of the corresponding face.

It follows from [33] that one can compute the Hasse-diagram of the  $k$ -skeleton (i.e., all faces of dimensions at most  $k$ ) of  $P$  in  $\mathcal{O}(n \cdot \alpha \cdot \varphi^{\leq k})$  time, where  $\varphi^{\leq k}$  is the number of faces of dimensions at most  $k$ . Since the latter number is in  $\mathcal{O}(n^{k+1})$ , the  $k$ -skeleton can be computed in polynomial time (in the input size) for fixed  $k$ .

**Related problems:** 5, 14

### 14. $f$ -VECTOR OF COMBINATORIAL POLYTOPES

**Input:** Vertex-facet incidence matrix of a polytope  $P$

**Output:**  $f$ -vector of  $P$

**Status (general):** Open

**Status (fixed dim.):** Polynomial time

By the remarks on Problem 13, it is clear that the first  $k$  entries of the  $f$ -vector can be computed in polynomial time for every fixed  $k$ .

If the polytope is simplicial and a shelling (or a partition) of its boundary complex is available (see Problems 17 and 18), then one can compute the entire  $f$ -vector in polynomial time [65, Chap. 8].

**Related problems:** 7, 13, 17, 18, 32

## 15. RECONSTRUCTION OF SIMPLE POLYTOPES

**Input:** The (abstract) graph  $\mathcal{G}_P$  of a simple polytope  $P$

**Output:** The family of the subsets of nodes of  $\mathcal{G}_P$  corresponding to the vertex sets of the facets of  $P$

**Status (general):** Open

**Status (fixed dim.):** Open

Blind and Mani [6] proved that the entire combinatorial structure of a simple polytope is determined by its graph. This is false for general polytopes (of dimension at least four), which is the main reason why we restrict our attention to simple polytopes for the remaining problems in this section. Kalai [35] gave a short, elegant, and constructive proof of Blind and Mani's result. However, the algorithm that can be derived from it has a worst-case running time that is exponential in the number of vertices of the polytope.

In [32] it is shown that the problem can be formulated as a combinatorial optimization problem for which the problem to find an AOF-orientation of  $\mathcal{G}_P$  (see Problem 17) is strongly dual in the sense of combinatorial optimization. In particular, the vertex sets of the facets of  $P$  have a *good characterization* in terms of  $\mathcal{G}_P$  (in the sense of Edmonds [18]). The problem is polynomial time equivalent to computing the cycles in  $\mathcal{G}_P$  that correspond to the 2-faces of  $P$ .

The problem can be solved in polynomial time in dimension at most three by computing a planar embedding of the graph, which can be done in linear time (Hopcroft and Tarjan [30], Mehlhorn and Mutzel [45]).

**Related problems:** 16, 17, 18

## 16. FACET SYSTEM VERIFICATION FOR SIMPLE POLYTOPES

**Input:** The (abstract) graph  $\mathcal{G}_P$  of a simple polytope  $P$  and a family  $\mathcal{F}$  of subsets of nodes of  $\mathcal{G}_P$

**Output:** “Yes” if  $\mathcal{F}$  is the family of subsets of nodes of  $\mathcal{G}_P$  that correspond to the vertex sets of the facets of  $P$ , “No” otherwise

**Status (general):** Open

**Status (fixed dim.):** Open

In [32] it is shown that both the “Yes”- as well as the “No”-answer can be proved in polynomial time in the size of  $\mathcal{G}_P$  (provided that the integrity of the input data is guaranteed). Any polynomial time algorithm for the construction of an AOF- or US-orientation (see Problems 17 and 18) of  $\mathcal{G}_P$  would yield a polynomial time algorithm for this problem (see [32]).

Up to dimension three the problem can be solved in polynomial time (see the comments to Problems 15 and 17).

**Related problems:** 15, 17, 18, 30

## 17. AOF-ORIENTATION

**Input:** The (abstract) graph  $\mathcal{G}_P$  of a simple polytope  $P$

**Output:** An AOF-orientation of  $\mathcal{G}_P$

**Status (general):** Open

**Status (fixed dim.):** Open

(Simple) polytopes admit AOF-orientations because every linear function in general position induces an AOF-orientation.

In [32] it is shown that one can formulate the problem as a combinatorial optimization problem, for which a strongly dual problem in the sense of combinatorial optimization exists (see the comments to Problem 15). Thus, the AOF-orientations of  $\mathcal{G}_P$  have a good characterization (see Problem 16) in terms of  $\mathcal{G}_P$ , i.e., there are polynomial size proofs for both cases an orientation being an AOF-orientation or not (provided that the integrity of the input data is guaranteed). However, it is unknown if it is possible to check in polynomial time if a given orientation is an AOF-orientation.

In dimensions one and two the problem is trivial. For a three-dimensional polytope  $P$  the problem can be solved in polynomial time, e.g., by producing a plane drawing of  $\mathcal{G}_P$  with convex faces (see Tutte [62]) and sorting the nodes with respect to a linear function (in general position).

A polynomial time algorithm would lead to a polynomial algorithm for Problem 16 (see [32]).

By duality of polytopes, the problem is equivalent to the problem of finding a shelling order of the facets of a simplicial polytope from the upper two layers of its face lattice. It is unknown whether it is possible to find in polynomial time a shelling order of the facets, even if the polytope is given by its entire face lattice. With this larger input, however, it is possible to check in polynomial time whether a given ordering of the facets is a shelling order.

**Related problems:** 16, 18, 34

## 18. US-ORIENTATION

**Input:** The (abstract) graph  $\mathcal{G}_P$  of a simple polytope  $P$

**Output:** A US-orientation of  $\mathcal{G}_P$

**Status (general):** Open

**Status (fixed dim.):** Open

Since AOF-orientations are US-orientations, it follows from the remarks on Problem 17 that (simple) polytopes admit US-orientations and that the problem can be solved in polynomial time up to dimension three. By slight adaptations of the arguments given in [32], one can prove that a polynomial time algorithm for this problem would yield a polynomial time algorithm for Problem 16 as well.

In contrast to Problem 17, no good characterization of US-orientations is known.

It is not hard to see that, by duality of polytopes, the problem is equivalent to the problem of finding from the upper two layers a partition of the face lattice of a simplicial polytope into intervals whose upper bounds are the facets (i.e., a partition in the sense of Stanley [58]). Similar to the situation with shelling orders, it is even unknown whether such a partition can be found in polynomial time if the polytope is specified by its entire face lattice. Again, with the entire face lattice as input it can be checked in polynomial time whether a family of subsets of the face lattice is a partition in that sense.

**Related problems:** 16, 17, 35

## 4 Isomorphism

Two polytopes  $P_1 \subset \mathbb{R}^{d_1}$  and  $P_2 \subset \mathbb{R}^{d_2}$  are *affinely equivalent* if there is a one-to-one affine map  $T : \text{aff}(P_1) \rightarrow \text{aff}(P_2)$  between the affine hulls of  $P_1$  and  $P_2$  with  $T(P_1) = P_2$ . Two polytopes are *combinatorially equivalent* (or *isomorphic*) if their face lattices are isomorphic. It is not hard to see that affine equivalence implies combinatorial equivalence.

As soon as one starts to investigate structural properties of polytopes by means of computer programs, algorithms for deciding whether two polytopes are isomorphic become relevant.

Some problems in this section are known to be hard in the sense that the *graph isomorphism problem* can polynomially be reduced to them. Although this problem is not known (and even not expected) to be  $\mathcal{NP}$ -complete, all attempts to find a polynomial time algorithm for it have failed so far. Actually, the same holds for a lot of problems that can be polynomially reduced to the graph isomorphism problem (see, e.g., Babai [3]).

### 19. AFFINE EQUIVALENCE OF $\mathcal{V}$ -POLYTOPES

**Input:** Two polytopes  $P$  and  $Q$  given in  $\mathcal{V}$ -description

**Output:** “Yes” if  $P$  is affinely equivalent to  $Q$ , “No” otherwise

**Status (general):** Graph isomorphism hard

**Status (fixed dim.):** Polynomial time

The graph isomorphism problem can polynomially be reduced to the problem of checking the affine equivalence of two polytopes [34]. The problem remains graph isomorphism hard if  $\mathcal{H}$ -descriptions are additionally provided as input data and/or if one restricts the input to simple or simplicial polytopes.

For polytopes of bounded dimension the problem can be solved in polynomial time by mere enumeration of affine bases among the vertex sets.

**Related problems:** 20

20. COMBINATORIAL EQUIVALENCE OF  $\mathcal{V}$ -POLYTOPES

**Input:** Two polytopes  $P$  and  $Q$  given in  $\mathcal{V}$ -description

**Output:** “Yes” if  $P$  is combinatorially equivalent to  $Q$ , “No” otherwise

**Status (general):**  $\text{co}\mathcal{NP}$ -hard

**Status (fixed dim.):** Polynomial time

Swart [60] describes a reduction of the subset-sum problem to the negation of the problem.

For polytopes of bounded dimension the problem can be solved in polynomial time (see Problems 2 and 22).

**Related problems:** 2, 19, 22

## 21. POLYTOPE ISOMORPHISM

**Input:** The face lattices  $\mathcal{L}_P$  and  $\mathcal{L}_Q$  of two polytopes  $P$  and  $Q$ , respectively

**Output:** “Yes” if  $\mathcal{L}_P$  is isomorphic to  $\mathcal{L}_Q$ , “No” otherwise

**Status (general):** Open

**Status (fixed dim.):** Polynomial time

The problem can be solved in polynomial time in constant dimension (see Problem 22). In general, the problem can easily be reduced to the graph isomorphism problem

**Related problems:** 22, 23

## 22. ISOMORPHISM OF VERTEX-FACET INCIDENCES

**Input:** Vertex-facet incidence matrices  $A_P$  and  $A_Q$  of polytopes  $P$  and  $Q$ , respectively

**Output:** “Yes” if  $A_P$  can be transformed into  $A_Q$  by row and column permutations, “No” otherwise

**Status (general):** Graph isomorphism complete

**Status (fixed dim.):** Polynomial time

The problem remains graph isomorphism complete even if  $\mathcal{V}$ - and  $\mathcal{H}$ -descriptions of  $P$  and  $Q$  are part of the input data [34].

In constant dimension the problem can be solved in polynomial time by a reduction [34] to the graph isomorphism problem for graphs of bounded degree, for which a polynomial time algorithm is known (Luks [41]).

Problem 21 can polynomially be reduced to this problem. For polytopes of bounded dimension both problems are polynomial time equivalent.

**Related problems:** 21, 20

### 23. SELFDUALITY OF POLYTOPES

**Input:** Face Lattice of a polytope  $P$

**Output:** “Yes” if  $P$  is isomorphic to its dual, “No” otherwise

**Status (general):** Open

**Status (fixed dim.):** Polynomial time

This is a special case of problem 21. In particular, it is solvable in polynomial time in bounded dimensions.

It is easy to see that deciding whether a general 0/1-matrix  $A$  (not necessarily a vertex-facet incidence matrix of a polytope) can be transformed into  $A^T$  by permuting its rows and columns is graph isomorphism complete.

**Related problems:** 21

## 5 Optimization

In this section, next to the original linear programming problem, we describe some of its relatives. In particular, combinatorial abstractions of the problem are important with respect to polytope theory (and, more general, discrete geometry). We pick out the aspect of combinatorial cube programming here (and leave aside abstractions like general combinatorial linear programming, LP-type problems, and oriented matroid programming), since it has received considerable attention lately.

### 24. GEOMETRIC LINEAR PROGRAMMING

**Input:**  $\mathcal{H}$ -description of a polyhedron  $P \subset \mathbb{Q}^d$ ,  $\mathbf{c} \in \mathbb{Q}^d$

**Output:**  $\inf \{\mathbf{c}^T \mathbf{x} \mid \mathbf{x} \in P\} \in \mathbb{Q} \cup \{-\infty, \infty\}$  and, if the infimum is finite, a point where the infimum is attained.

**Status (general):** Polynomial time; no *strongly* polynomial time algorithm known

**Status (fixed dim.):** Linear time in  $m$  (the number of inequalities)

The first polynomial time algorithm was a variant of the *ellipsoid algorithm* due to Khachiyan [38]. Later, also *interior point methods* solving the problem in polynomial time were discovered (Karmarkar [37]).

Megiddo found an algorithm solving the problem for a fixed number  $d$  of variables in  $\mathcal{O}(m)$  arithmetic operations (Megiddo [44]).

No strongly polynomial time algorithm (performing a number of arithmetic operations that is bounded polynomially in  $d$  and the number of half-spaces rather than in the coding lengths of the input coordinates) is known. In particular, no polynomial time variant of the *simplex algorithm* is known. However, a randomized version of the simplex algorithm solves the problem in (expected) subexponential time (Kalai [36], Matoušek, Sharir, and Welzl [42]).

**Related problems:** 25, 26, 27

## 25. OPTIMAL VERTEX

**Input:**  $\mathcal{H}$ -description of a polyhedron  $P \subset \mathbb{Q}^d$ ,  $\mathbf{c} \in \mathbb{Q}^d$

**Output:**  $\inf \{\mathbf{c}^T \mathbf{v} \mid \mathbf{v} \text{ vertex of } P\} \in \mathbb{Q} \cup \{\infty\}$  and, if the infimum is finite, a vertex where the infimum is attained.

**Status (general):** Strongly  $\mathcal{NP}$ -hard

**Status (fixed dim.):** Polynomial time

Proved to be strongly  $\mathcal{NP}$ -hard by Fukuda, Liebling, and Margot [22]. By linear programming this problem can be solved in polynomial time if  $P$  is a polytope. In fixed dimension one can compute all vertices of  $P$  in polynomial time (see Problem 1).

**Related problems:** 1, 24, 26

## 26. VERTEX WITH SPECIFIED OBJECTIVE VALUE

**Input:**  $\mathcal{H}$ -description of a polyhedron  $P \subset \mathbb{Q}^d$ ,  $\mathbf{c} \in \mathbb{Q}^d$ ,  $C \in \mathbb{Q}$

**Output:** “Yes” if there is a vertex  $\mathbf{v}$  of  $P$  with  $\mathbf{c}^T \mathbf{v} = C$ ; “No” otherwise

**Status (general):** Strongly  $\mathcal{NP}$ -complete

**Status (fixed dim.):** Polynomial time

Proved to be  $\mathcal{NP}$ -complete by Chandrasekaran, Kabadi, and Murty [11] and strongly  $\mathcal{NP}$ -complete by Fukuda, Liebling, and Margot [22]. The problem remains strongly  $\mathcal{NP}$ -complete even if the input is restricted to polytopes [22].

**Related problems:** 24, 25

## 27. AOF CUBE PROGRAMMING

**Input:** An oracle for a function  $\sigma : \{0, 1\}^d \rightarrow \{+, -\}^d$  defining an AOF-orientation of the graph of the  $d$ -cube

**Output:** The sink of the orientation

**Status (general):** Open

**Status (fixed dim.):** Constant time

The problem can be solved in a subexponential number of oracle calls by the *random facet* variant of the simplex algorithm due to Kalai [36]. For a derivation of the explicit bound  $e^{2\sqrt{d}} - 1$  see Gärtner [25].

In fixed dimension the problem is trivial by mere enumeration.

The problem generalizes linear programming problems whose sets of feasible solutions are combinatorially equivalent to cubes.

**Related problems:** 24, 28



## 28. USO CUBE PROGRAMMING

**Input:** An oracle for a function  $\sigma : \{0, 1\}^d \rightarrow \{+, -\}^d$  defining a US-orientation of the graph of the  $d$ -cube

**Output:** The sink of the orientation

**Status (general):** Open

**Status (fixed dim.):** Constant time

Szabó and Welzl [61] describe a randomized algorithm solving the problem in an expected number of  $\mathcal{O}(\alpha^d)$  oracle calls with  $\alpha = \sqrt{43/20} < 1.467$  and a deterministic algorithm that needs  $\mathcal{O}(1.61^d)$  oracle calls. Plugging an optimal algorithm for the three-dimensional case (found by Günter Rote) into their algorithm, Szabó and Welzl even obtain an  $\mathcal{O}(1.438^d)$  randomized algorithm. The problem not only generalizes Problem 27, but also certain linear complementary problems and smallest enclosing ball problems.

In fixed dimension the problem is trivial by mere enumeration.

**Related problems:** 27

## 6 Realizability

In this section problems are discussed which bridge the gap from combinatorial descriptions of polytopes to geometrical descriptions, i.e., it deals with questions of the following kind: given combinatorial data, does there exist a polytope which “realizes” this data? E.g., given a 0/1-matrix is this the matrix of vertex-facet incidences of a polytope? The problems of computing combinatorial from geometrical data is discussed in Section 2.

The problems listed in this section are among the first ones asked in (modern) polytope theory, going back to the work of Steinitz and Radermacher in the 1930’s [59].

## 29. STEINITZ PROBLEM

**Input:** Lattice  $\mathcal{L}$

**Output:** “Yes” if  $\mathcal{L}$  is isomorphic to the face lattice of a polytope, “No” otherwise

**Status (general):**  $\mathcal{NP}$ -hard

**Status (fixed dim.):**  $\mathcal{NP}$ -hard

If  $\mathcal{L}$  is isomorphic to the face lattice of a polytope, it is ranked, atomic, and coatomic. These properties can be tested in polynomial time in the size of  $\mathcal{L}$ . Furthermore, in this case, the dimension  $d$  of a candidate polytope has to be  $\text{rank } \mathcal{L} - 1$ .

The problem is trivial for dimension  $d \leq 2$ . Steinitz’s Theorem allows to solve  $d = 3$  in polynomial time: construct the (abstract) graph  $G$ , test if the facets

can consistently be embedded in the plane (linear time [30,45]) and check for 3-connectedness (in linear time, see Hopcroft and Tarjan [29]).

Mněv proved that the Steinitz Problem for  $d$ -polytopes with  $d + 4$  vertices is  $\mathcal{NP}$ -hard [47]. Even more, Richter-Gebert [53] proved that for (fixed)  $d \geq 4$  the problem is  $\mathcal{NP}$ -hard.

For fixed  $d \geq 4$  it is neither known whether the problem is in  $\mathcal{NP}$  nor whether it is in  $\text{co}\mathcal{NP}$ . It seems unlikely to be in  $\mathcal{NP}$ , since there are 4-polytopes which cannot be realized with rational coordinates of coding length which is bounded by a polynomial in  $|\mathcal{L}|$  (see Richter-Gebert [53]).

**Related problems:** 30

### 30. SIMPLICIAL STEINITZ PROBLEM

**Input:** Lattice  $\mathcal{L}$

**Output:** “Yes” if  $\mathcal{L}$  is isomorphic to the face lattice of a simplicial polytope, “No” otherwise

**Status (general):**  $\mathcal{NP}$ -hard

**Status (fixed dim.):** Open

As for Problem 29,  $\mathcal{L}$  is ranked, atomic, and coatomic if the answer is “Yes.” In this case, the dimension  $d$  of any matched polytope is  $\text{rank } \mathcal{L} - 1$ .

As for general polytopes (Problem 29), this problem is polynomial time solvable in dimension  $d \leq 3$ .

The problem is  $\mathcal{NP}$ -hard, which follows from the above mentioned fact that the Steinitz problem for  $d$ -polytopes with  $d + 4$  vertices is  $\mathcal{NP}$ -hard and a construction (Bokowski and Sturmfels [7]) which generalizes it to the *simplicial* case (but increases the dimension). It is, however, open whether the problem is  $\mathcal{NP}$ -hard for *fixed* dimension. For fixed  $d \geq 4$ , it is neither known whether the problem is in  $\mathcal{NP}$  nor whether it is in  $\text{co}\mathcal{NP}$ .

The following question is interesting in connection with Problem 16 (see also the notes there): Given an (abstract) graph  $G$ , is  $G$  the graph of a simple polytope? If we do not restrict the question to simple polytopes the problem is also interesting.

**Related problems:** 16, 29

## 7 Beyond Polytopes

This section is concerned with problems on finite abstract simplicial complexes. Some of the problems listed are direct generalizations of problems on polytopes. Most of the basic notions relevant in our context can be looked up in [65]; for topological concepts like *homology* we refer to Munkres’ book [48].

A *finite abstract simplicial complex*  $\Delta$  is a non-empty set of subsets (the *simplices* or *faces*) of a finite set of *vertices* such that  $F \in \Delta$  and  $G \subset F$  imply

$G \in \Delta$ . The *dimension* of a simplex  $F \in \Delta$  is  $|F| - 1$ . The *dimension*  $\dim(\Delta)$  of  $\Delta$  is the largest dimension of any of the simplices in  $\Delta$ . If all its maximal simplices with respect to inclusion (i.e., its *facets*) have the same cardinality, then  $\Delta$  is *pure*. A pure  $d$ -dimensional finite abstract simplicial complex whose *dual graph* (defined on the facets, where two facets are adjacent if they share a common  $(d - 1)$ -face) is connected is a *pseudo-manifold* if every  $(d - 1)$ -dimensional simplex is contained in at most two facets. The boundary of a simplicial  $(d + 1)$ -dimensional polytope induces a  $d$ -dimensional pseudo-manifold.

Throughout this section a finite abstract simplicial complex  $\Delta$  is given by its list of facets or by the complete list of all simplices. In the first case, the input size can be measured by  $n$  and  $m$ , the numbers of vertices and facets.

### 31. EULER CHARACTERISTIC

**Input:** Finite abstract simplicial complex  $\Delta$  given by a list of facets

**Output:** Euler characteristic  $\chi(\Delta) \in \mathbb{Z}$

**Status (general):** Open

**Status (fixed dim.):** Polynomial time

It is unknown whether the decision version “ $\chi(\Delta) = 0$ ?” of this problem is in  $\mathcal{NP}$ . The problem is easy if  $\Delta$  is given by a list of all of its simplices. For fixed dimension, one can enumerate all simplices of  $\Delta$  and compute the Euler characteristic in polynomial time.

Currently the fastest way to compute the Euler characteristic is to determine  $\mathcal{V} = \{S : S \text{ is an intersection of facets of } \Delta\}$  and then compute  $\chi(\Delta)$  in time  $\mathcal{O}(|\mathcal{V}|^2)$  by a Möbius function approach, see Rota [54]. Usually  $\mathcal{V}$  is much smaller than the whole face lattice of  $\Delta$ .  $\mathcal{V}$  can be listed lexicographically by an algorithm of Ganter [23], in time  $\mathcal{O}(\min\{m, n\} \cdot \alpha \cdot |\mathcal{V}|)$ , where  $\alpha$  is the number of vertex-facets incidences.

**Related problems:** 32

### 32. $f$ -VECTOR OF SIMPLICIAL COMPLEXES

**Input:** Finite abstract simplicial complex  $\Delta$  given by a list of facets

**Output:** The  $f$ -vector of  $\Delta$

**Status (general):**  $\#\mathcal{P}$ -hard

**Status (fixed dim.):** Polynomial time

If  $\Delta$  is given by all of its simplices the problem is trivial. Clearly, for fixed  $k$ , the first  $k$  entries of the  $f$ -vector can be computed in polynomial time, since the number of  $k$ -simplices in  $\Delta$  is polynomial in  $n$ . Hence the problem is polynomial time solvable for fixed dimension  $\dim(\Delta)$ .

It is unknown whether the decision problem “Given the list of facets of  $\Delta$  and some  $\varphi \in \mathbb{N}$ ; is  $\varphi$  the total number of faces of  $\Delta$ ?” is contained in  $\mathcal{NP}$ .

This problem is only known to be in  $\mathcal{NP}$  for partitionable (see Problem 18) simplicial complexes (see Kleinschmidt and Onn [39]).

To the best of our knowledge, no proof of  $\#\mathcal{P}$ -hardness of the general problem has appeared in the literature. Therefore we include one here.

Consider an instance of SAT, i.e., a formula in conjunctive normal form (CNF-formula)  $C_1 \wedge \dots \wedge C_m$  with variables  $x_1, \dots, x_n$  (each  $C_i$  contains only disjunctions of literals). It is well known (Valiant [63]) that computing the number of satisfying truth assignments is  $\#\mathcal{P}$ -complete. Define  $E = \{t_1, f_1, \dots, t_n, f_n\}$ . Part I. First, let  $E$  be the vertex set of a simplicial complex  $\Delta$  defined by the minimal non-faces (*circuits*)  $C'_1, \dots, C'_m, P_1, \dots, P_n$ , where  $P_i = \{t_i, f_i\}$  for every  $i$ . Here for any clause  $C$ ,  $C' := \{f_j : x_j \text{ literal in } C\} \cup \{t_j : \bar{x}_j \text{ literal in } C\}$ , e.g., for  $C = x_1 \vee x_2 \vee \bar{x}_3$  we have  $C' = \{f_1, f_2, t_3\}$ . The idea is that  $t_i$  corresponds to the assignment of a true-value and  $f_i$  corresponds to the assignment of a false-value to variable  $x_i$ . The circuits exclude subsets of  $E$  which include both  $t_i$  and  $f_i$  for all variables  $x_i$  and exclude truth-assignments to variables which would not satisfy a clause  $C_j$ . It is, however, allowed that for some variable  $x_i$  neither  $t_i$  nor  $f_i$  is included in a face. But every  $(n-1)$ -face ( $n$ -subset of  $E$ ) (if there exists any) corresponds to a truth-assignment to the variables (which uses exactly one value for each variable) and satisfies the formula. These subsets are counted by  $f_{n-1}(\Delta)$ . Hence computing  $f_{n-1}$  is  $\#\mathcal{P}$ -complete and computing the  $f$ -vector of  $\Delta$  is  $\#\mathcal{P}$ -hard. Moreover this shows that computing the dimension of a simplicial complex given by the minimal non-faces is  $\mathcal{NP}$ -hard.

Part II. We now construct a simplicial complex  $\overline{\Delta}$  (the *dual complex*) which is given by facets. Define  $\overline{\Delta}$  by the facets  $\overline{C}'_1, \dots, \overline{C}'_m, \overline{P}_1, \dots, \overline{P}_n$ , where for  $S \subseteq E$ ,  $\overline{S} := E \setminus S$ . We have that a set  $S \subseteq E$  is a face of  $\Delta$  if and only if  $\overline{S}$  is *not* a face of  $\overline{\Delta}$ . Hence,  $f_{n-1}(\Delta) + f_{n-1}(\overline{\Delta}) = \binom{2n}{n}$ . It follows that one can efficiently compute  $f_{n-1}(\Delta)$  from  $f_{n-1}(\overline{\Delta})$ .

**Related problems:** 14, 31

### 33. HOMOLOGY

**Input:** Finite abstract simplicial complex  $\Delta$  given by a list of facets,  $i \in \mathbb{N}$

**Output:** The  $i$ -th homology group of  $\Delta$ , given by its rank and its torsion coefficients

**Status (general):** Open

**Status (fixed dim.):** Polynomial time

There exists a polynomial time algorithm if  $\Delta$  is given by the list of all simplices, since the Smith normal form of an integer matrix can be computed efficiently (Iliopoulos [31]). For fixed  $i$  or  $\dim(\Delta) - i$ , the sizes of the boundary matrices are polynomial in the size of  $\Delta$  and the Smith normal form can again be computed efficiently.

**Related problems:** 31, 32

## 34. SHELLABILITY

**Input:** Finite abstract pure simplicial complex  $\Delta$  given by a list of facets

**Output:** “Yes” if  $\Delta$  is shellable, “No” otherwise

**Status (general):** Open

**Status (fixed dim.):** Open

Given an ordering of the facets of  $\Delta$ , it can be tested in polynomial time whether it is a shelling order. Hence, the problem is in  $\mathcal{NP}$ .

The problem can be solved in polynomial time for one-dimensional complexes, i.e., for graphs: a graph is shellable if and only if it is connected. Even for  $\dim(\Delta) = 2$ , the status is open. In particular, it is unclear if the problem can be solved in polynomial time if  $\Delta$  is given by a list of all simplices.

For two-dimensional pseudo-manifolds the problem can be solved in linear time (Danaraj and Klee [13]).

**Related problems:** 17, 35

## 35. PARTITIONABILITY

**Input:** Finite abstract simplicial complex  $\Delta$  given by a list of facets

**Output:** “Yes” if  $\Delta$  is partitionable, “No” otherwise

**Status (general):** Open

**Status (fixed dim.):** Open

As in Problem 18, partitionability is meant in the sense of Stanley [58] (see also [65]). Noble [50] proved that the problem is in  $\mathcal{NP}$ .

PARTITIONABILITY can be solved in polynomial time for one-dimensional complexes, i.e., for graphs: a graph is partitionable if and only if at most one of its connected components is a tree. For two-dimensional complexes the complexity status is open. In particular, it is unclear if the problem can be solved in polynomial time if  $\Delta$  is given by a list of all simplices.

**Related problems:** 18, 34

**Table of Problems**

Coordinate Descriptions .....	4
1. VERTEX ENUMERATION .....	4
2. FACET ENUMERATION .....	5
3. POLYTOPE VERIFICATION .....	5
4. POLYTOPE CONTAINMENT .....	6
5. FACE LATTICE OF GEOMETRIC POLYTOPES .....	6
6. DEGENERACY TESTING .....	6
7. NUMBER OF VERTICES .....	7
8. FEASIBLE BASIS EXTENSION .....	7
9. RECOGNIZING INTEGER POLYHEDRA .....	8
10. DIAMETER .....	8
11. MINIMUM TRIANGULATION .....	8
12. VOLUME .....	9
Combinatorial Structure .....	9
13. FACE LATTICE OF COMBINATORIAL POLYTOPES .....	10
14. $f$ -VECTOR OF COMBINATORIAL POLYTOPES .....	10
15. RECONSTRUCTION OF SIMPLE POLYTOPES .....	11
16. FACET SYSTEM VERIFICATION FOR SIMPLE POLYTOPES .....	11
17. AOF-ORIENTATION .....	12
18. US-ORIENTATION .....	12
Isomorphism .....	13
19. AFFINE EQUIVALENCE OF $\mathcal{V}$ -POLYTOPES .....	13
20. COMBINATORIAL EQUIVALENCE OF $\mathcal{V}$ -POLYTOPES .....	14
21. POLYTOPE ISOMORPHISM .....	14
22. ISOMORPHISM OF VERTEX-FACET INCIDENCES .....	14
23. SELF DUALITY OF POLYTOPES .....	15
Optimization .....	15
24. GEOMETRIC LINEAR PROGRAMMING .....	15
25. OPTIMAL VERTEX .....	16
26. VERTEX WITH SPECIFIED OBJECTIVE VALUE .....	16
27. AOF CUBE PROGRAMMING .....	16
28. USO CUBE PROGRAMMING .....	17
Realizability .....	17
29. STEINITZ PROBLEM .....	17
30. SIMPLICIAL STEINITZ PROBLEM .....	18
Beyond Polytopes .....	18
31. EULER CHARACTERISTIC .....	19
32. $f$ -VECTOR OF SIMPLICIAL COMPLEXES .....	19
33. HOMOLOGY .....	20
34. SHELLABILITY .....	21
35. PARTITIONABILITY .....	21

## References

1. D. AVIS, D. BREMNER, AND R. SEIDEL, *How good are convex hull algorithms?*, *Comput. Geom.*, 7 (1997), pp. 265–301.
2. D. AVIS AND K. FUKUDA, *A pivoting algorithm for convex hull and vertex enumeration of arrangements and polyhedra*, *Discrete Comput. Geom.*, 8 (1992), pp. 295–313.
3. L. BABAI, *Automorphism groups, isomorphism, reconstruction*, in *Handbook of Combinatorics*, R. L. Graham, M. Grötschel, and L. Lovász, eds., vol. 2, Elsevier (North-Holland), Amsterdam, 1995, pp. 1447–1540.
4. A. BELOW, J. A. DE LOERA, AND J. RICHTER-GEBERT, *The complexity of finding small triangulations of convex 3-polytopes*. *Journal of Algorithms*, to appear; arXiv:math.CO/0012177, 2000.
5. ———, *Finding minimal triangulations of convex 3-polytopes is NP-hard*, in *Proceedings of the 11th annual ACM-SIAM symposium on Discrete algorithms*, San Francisco, CA, USA, SIAM, Philadelphia, 2000, pp. 65–66.
6. R. BLIND AND P. MANI-LEVITSKA, *On puzzles and polytope isomorphisms*, *Aequationes Math.*, 34 (1987), pp. 287–297.
7. J. BOKOWSKI AND B. STURMFELS, *Computational Synthetic Geometry*, no. 1355 in *Lecture Notes in Mathematics*, Springer-Verlag, 1989.
8. D. BREMNER, K. FUKUDA, AND A. MARZETTA, *Primal-dual methods for vertex and facet enumeration*, *Discrete Comput. Geom.*, 20 (1998), pp. 333–357.
9. T. M. CHAN, *Output-sensitive results on convex hulls, extreme points, and related problems*, *Discrete Comput. Geom.*, 16 (1996), pp. 369–387.
10. D. R. CHAND AND S. S. KAPUR, *An algorithm for convex polytopes*, *J. Assoc. Comput. Mach.*, 17 (1970).
11. R. CHANDRASEKARAN, S. N. KABADI, AND K. G. MURTY, *Some NP-complete problems in linear programming*, *Oper. Res. Lett.*, 1 (1982), pp. 101–104.
12. B. CHAZELLE, *An optimal convex hull algorithm in any fixed dimension*, *Discrete Comput. Geom.*, 10 (1993).
13. G. DANARAJ AND V. KLEE, *A representation of 2-dimensional pseudomanifolds and its use in the design of a linear-time shelling algorithm.*, *Ann. Discrete Math.*, 2 (1978), pp. 53–63.
14. M. E. DYER, *The complexity of vertex enumeration methods*, *Math. Oper. Res.*, 8 (1983), pp. 381–402.
15. M. E. DYER AND A. M. FRIEZE, *On the complexity of computing the volume of a polyhedron.*, *SIAM J. Comput.*, 17 (1988), pp. 967–974.
16. M. E. DYER, A. M. FRIEZE, AND R. KANNAN, *A random polynomial-time algorithm for approximating the volume of convex bodies.*, *J. Assoc. Comput. Mach.*, 38 (1991), pp. 1–17.
17. B. C. EAVES AND R. M. FREUND, *Optimal scaling of balls and polyhedra*, *Math. Program.*, 23 (1982), pp. 138–147.
18. J. EDMONDS, *Paths, trees, and flowers*, *Can. J. Math.*, 17 (1965), pp. 449–467.
19. J. ERICKSON, *New lower bounds for convex hull problems in odd dimensions*, *SIAM J. Comput.*, 28 (1999), pp. 1198–1214.
20. R. M. FREUND AND J. B. ORLIN, *On the complexity of four polyhedral set containment problems*, *Math. Program.*, 33 (1985), pp. 139–145.
21. A. M. FRIEZE AND S.-H. TENG, *On the complexity of computing the diameter of a polytope*, *Comput. Complexity*, 4 (1994), pp. 207–219.

22. K. FUKUDA, T. M. LIEBLING, AND F. MARGOT, *Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron*, *Comput. Geom.*, 8 (1997), pp. 1–12.
23. B. GANTER, *Algorithmen zur Formalen Begriffsanalyse*, in *Beiträge zur Begriffsanalyse*, B. Ganter, R. Wille, and K. E. Wolff, eds., B.I. Wissenschaftsverlag, 1987, pp. 241–254.
24. M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of  $\mathcal{NP}$ -Completeness*, W. H. Freeman and Company, San Francisco, 1979.
25. B. GÄRTNER, *The random facet simplex algorithm on combinatorial cubes*. Manuscript, February 2001.
26. E. GAWRILOW AND M. JOSWIG, *Homepage of `polymake`*. <http://www.math.tu-berlin.de/diskregeom/polymake/>.
27. E. GAWRILOW AND M. JOSWIG, *`polymake`: A framework for analyzing convex polytopes*, in *Polytopes — Combinatorics and Computation*, G. Kalai and G. M. Ziegler, eds., Birkhäuser, 2000, pp. 43–74.
28. ———, *`polymake`: an approach to modular software design in computational geometry*, in *Proc. 17th ACM Ann. Symp. Comput. Geom.*, 2001, pp. 222–231.
29. J. E. HOPCROFT AND R. E. TARJAN, *Dividing a graph into triconnected components*, *SIAM J. Comput.*, 2 (1973), pp. 135–158.
30. ———, *Efficient planarity testing*, *J. Assoc. Comput. Mach.*, 21 (1974), pp. 549–568.
31. C. S. ILIOPOULOS, *Worst-case complexity bounds on algorithms for computing the canonical structure of finite Abelian groups and the Hermite and Smith normal forms of an integer matrix*, *SIAM J. Comput.*, 18 (1989), pp. 658–669.
32. M. JOSWIG, V. KAIBEL, AND F. KÖRNER, *On the  $k$ -systems of a simple polytope*. To appear in: *Isr. J. Math.*
33. V. KAIBEL AND M. E. PFETSCH, *Computing the face lattice of a polytope from its vertex-facet incidences*, tech. report, TU Berlin, 2001. Submitted.
34. V. KAIBEL AND A. SCHWARTZ, *On the complexity of isomorphism problems related to polytopes*, tech. report, TU Berlin, 2001. Submitted.
35. G. KALAI, *A simple way to tell a simple polytope from its graph*, *J. Comb. Theory, Ser. A*, 49 (1988), pp. 381–383.
36. G. KALAI, *A subexponential randomized simplex algorithm*, in *Proc. 24th Ann. ACM Symp. Theory Comput.*, Victoria, 1992, ACM Press, pp. 475–482.
37. N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, *Combinatorica*, 4 (1984), pp. 373–395.
38. L. KHACHIYAN, *A polynomial algorithm in linear programming*, *Sov. Math., Dokl.*, 20 (1979), pp. 191–194.
39. P. KLEINSCHMIDT AND S. ONN, *Signable posets and partitionable simplicial complexes*, *Discrete Comput. Geom.*, 15 (1996), pp. 443–466.
40. N. LINIAL, *Hard enumeration problems in geometry and combinatorics*, *SIAM J. Alg. Disc. Math.*, 7 (1986), pp. 331–335.
41. E. M. LUKS, *Isomorphism of graphs of bounded valence can be tested in polynomial time*, *J. Comput. Syst. Sci.*, 25 (1982), pp. 42–65.
42. J. MATOUŠEK, M. SHARIR, AND E. WELZL, *A subexponential bound for linear programming*, *Algorithmica*, 16 (1996), pp. 498–516.
43. P. MCMULLEN, *The maximum numbers of faces of a convex polytope*, *Mathematika*, 17 (1970), pp. 179–184.



44. N. MEGIDDO, *Linear programming in linear time when the dimension is fixed*, J. Assoc. Comput. Mach., 31 (1984), pp. 114–127.
45. K. MEHLHORN AND P. MUTZEL, *On the embedding phase of the Hopcroft and Tarjan planarity testing algorithm*, Algorithmica, 16 (1996), pp. 233–242.
46. H. MINKOWSKI, *Geometrie der Zahlen (Geometry of numbers)*. Teubner Verlag, Leipzig, 1886 and 1910; reprinted by Chelsea, New York, 1953, and by Johnson, New York, 1968. (German).
47. N. E. MNĚV, *The universality theorems on the classification problem of configuration varieties and convex polytopes varieties*, in Topology and Geometry – Rohlin Seminar, O. Y. Viro, ed., no. 1346 in Lecture Notes in Mathematics, Springer-Verlag, 1988, pp. 527–543.
48. J. R. MUNKRES, *Elements of Algebraic Topology*, Addison-Wesley, Menlo Park CA, 1984.
49. K. G. MURTY, *A fundamental problem in linear inequalities with applications to the travelling salesman problem*, Math. Program., 2 (1972), pp. 296–308.
50. S. D. NOBLE, *Recognising a partitionable simplicial complex is in  $\mathcal{NP}$* , Discrete Math., 152 (1996), pp. 303–305.
51. C. H. PAPADIMITRIOU AND M. YANNAKAKIS, *On recognizing integer polyhedra*, Combinatorica, 10 (1990), pp. 107–109.
52. J. S. PROVAN, *Efficient enumeration of the vertices of polyhedra associated with network LP's*, Math. Program., 63 (1994), pp. 47–64.
53. J. RICHTER-GEBERT, *Realization Spaces of Polytopes*, no. 1643 in Lecture Notes in Mathematics, Springer-Verlag, 1996.
54. G.-C. ROTA, *On the foundations of combinatorial theory – I. Theory of Möbius functions*, Z. Wahrscheinlichkeitstheorie, 2 (1964), pp. 340–368.
55. A. SCHRIJVER, *Theory of linear and integer programming*, Wiley-Interscience, 1986.
56. R. SEIDEL, *Constructing higher-dimensional convex hulls at logarithmic cost per face*, in Proc. 18th Ann. ACM Sympos. Theory Comput., 1986, pp. 404–413.
57. ———, *Convex hull computations*, in Handbook of Discrete and Computational Geometry, J. Goodman and J. O'Rourke, eds., CRC Press, Boca Raton, 1997, ch. 19.
58. R. P. STANLEY, *Balanced Cohen-Macaulay complexes*, Trans. Am. Math. Soc., 249 (1979), pp. 139–157.
59. E. STEINITZ AND H. RADEMACHER, *Vorlesungen über die Theorie der Polyeder*, Springer Verlag, 1934. Reprint 1976.
60. G. F. SWART, *Finding the convex hull facet by facet*, J. Algorithms, 6 (1985), pp. 17–48.
61. T. SZABÓ AND E. WELZL, *Unique sink orientations of cubes*, tech. report, ETH Zürich, 2001. To appear in: Proc. 42nd Ann. Sympos. Found. Computer Science, Las Vegas, Oct 14–17, 2001.
62. W. T. TUTTE, *How to draw a graph*, Proc. Lond. Math. Soc., III. Ser., 13 (1963), pp. 743–768.
63. L. G. VALIANT, *The complexity of enumeration and reliability problems*, SIAM J. Comput., 8 (1979), pp. 410–421.
64. H. WEYL, *Elementare Theorie der konvexen Polyeder*, Comment. Math. Helv., 7 (1935), pp. 290–306.
65. G. M. ZIEGLER, *Lectures on Polytopes*, Springer-Verlag, 1995. Revised edition 1998.