

Convex Hulls, Oracles, and Homology*

Michael Joswig⁺ Günter M. Ziegler⁺

Inst. Mathematics, MA 6-2
TU Berlin, D-10623 Berlin, Germany
{joswig,ziegler}@math.tu-berlin.de

January 10, 2003

Abstract

This paper presents a new algorithm for the convex hull problem, which is based on a reduction to a combinatorial decision problem `COMPLETENESSC`, which in turn can be solved by a simplicial homology computation. Like other convex hull algorithms, our algorithm is polynomial (in the size of input plus output) for simplicial or simple input. We show that the “no”-case of `COMPLETENESSC` has a certificate that can be checked in polynomial time (if integrity of the input is guaranteed).

1 Introduction

Every convex polytope $P \subset \mathbb{R}^d$ can be described as the convex hull of a finite set \mathcal{P} of points or as the (bounded) set of solutions of a finite system \mathcal{H} of linear equations and inequalities [23, Lect. 1]. In view of the fundamental role that polytopes play in Euclidean geometry and hence for any type of geometric computing, the conversion between the two types of representations, known as the *convex hull problem*, is of key interest. It splits into two separate tasks.

The first task is the *facet enumeration problem*: Given a finite set of points $\mathcal{P} \subset \mathbb{R}^d$, determine the combinatorial structure of its boundary. For this one does not want to explicitly enumerate all the faces (the intersections of P with supporting hyperplanes), but one wants sparser data, namely to compute a minimal representation of the convex hull $\text{conv}(\mathcal{P})$ in terms of equations and (facet-defining) inequalities. Here the equations should describe the affine hull $\text{aff}(P)$, while the additional inequalities correspond to the *facets* (faces of codimension 1) of P . If P is full-dimensional in \mathbb{R}^d , then the facet-defining inequalities are unique up to scaling.

The second task is the *vertex enumeration problem*: Given a finite system \mathcal{H} of linear (equations and) inequalities, and provided that the set of solutions $P = \bigcap \mathcal{H}$ is bounded, compute the minimal set of points \mathcal{P} whose convex hull is P . This minimal set is unique; it consists of the *vertices* (0-dimensional faces) of P .

*An extended abstract version of this paper, “Polytope verification by homology verification,” has appeared in the Proceedings of EuroCG, Berlin, March 26–28, 2001, pp. 142–145.

⁺Partially supported by Deutsche Forschungs-Gemeinschaft (DFG)

The two tasks are dual to each other, via cone polarity. Thus if an LP-type oracle (an algorithm which for a system of inequalities computes a solution, or for a set of points computes a separating hyperplane, cf. [14]) is available, every algorithm for the facet enumeration problem can also be used for vertex enumeration, and vice versa.

Despite the great interest in the convex hull problem, and despite the fact that a number of different strategies and algorithms have been explored, implemented and analyzed in detail (see [10], as well as Avis [1] [2], Fukuda [11] and Gawrilow & Joswig [12] [13] for implementations), the problem can be considered “solved” neither in theory, nor in practice. If the dimension d is fixed, Chazelle’s celebrated algorithm [8] gives an asymptotically worst-case optimal (polynomial time) theoretical solution. Its optimality is based on McMullen’s “Upper Bound Theorem” [20] on the maximal number of facets for a d -polytope with n vertices. However, for any given convex hull problem, the output may be small, but it may also be much larger than the input — indeed, it may be of exponential size, if the dimension is not fixed. This is very relevant, since high-dimensional computations occur in a variety of important applications. Thus one is asking for a convex hull algorithm whose running time is bounded by a polynomial in the size of “input plus output”? Such an algorithm would be called *output-sensitive*. The analysis by Avis, Bremner and Seidel [3] shows that, unfortunately, none of the known types of convex hull algorithms is output-sensitive. These can roughly be categorized as follows: Incremental and triangulation producing (e.g., Chazelle’s method), incremental without triangulations (e.g., Fourier-Motzkin elimination [23, Lect. 1]), non-incremental (e.g., reverse search [4]). Note that, by a result of Bremner [6], only non-incremental methods can possibly be output-sensitive.

The purpose of this paper is to describe a new (non-incremental) convex hull algorithm, based on a completely different principle. To this end, we first present a (folklore) polynomial reduction of FACETENUMERATION to the decision problem POLYTOPEVERIFICATION. Then we further reduce to the COMPLETENESS problem: Is a given description of a d -polytope by *some* of its vertices and *some* of its facets complete, that is, are we given *all* the vertices and *all* the facets? Looking at the convex hull problem via its reduction to POLYTOPEVERIFICATION or COMPLETENESS automatically reveals its inherent self-dual structure. It is an interesting feature that the COMPLETENESS problem can be posed both with geometric input data and as an entirely combinatorial problem COMPLETENESSC, where only the incidences between vertices and facets are given.

Let us just mention here one recent occurrence of the combinatorial completeness problem: McCarthy et al. [19] describe a situation where one wants to know whether a given inequality description for a polytope is complete. Moreover, the vertex coordinates in some of their problems are necessarily non-rational, so any coordinate-free/combinatorial approach is welcome. Unfortunately, the most interesting case left “open” by McCarthy et al. (the convex hull of the matrices corresponding to the Coxeter group H_4) is a polytope completeness problem in dimension $d = 16$ with 14,400 vertices: From this data our method generates gigantic boundary matrices that are plainly too large to process.

Also we have been informed by Samuel Fiorini (email, January 2002) that he has successfully used a certificate for the “no”-case of COMPLETENESSC that is similar to the one that we describe in Section 6.

Our main contribution is an algorithm to attack the combinatorial COMPLETENESSC

problem via deciding whether a certain simplicial homology group of a certain abstract simplicial complex vanishes or not. Moreover, we present a polynomially checkable certificate for non-completeness, provided that the input is valid. For the geometric version the validity of the input can be checked easily. Unfortunately, the complexity status for the homology computation problem is open. The best currently available strategy to decide non-triviality of a (rational) homology group in question seems to be to compute boundary matrices and perform Gauss elimination. Since the boundary matrices in our algorithm can be exponentially large, we do not obtain an output-sensitive method. However, like other methods (e.g., Avis' and Fukuda's reverse search [4] or Seidel's gift-wrapping algorithm [22]) our algorithm is output-sensitive in the case of simplicial polytopes.

2 FacetEnumeration via PolytopeVerification

We start with a more formal description of the facet enumeration problem:

FACETENUMERATION(e, \mathcal{P}):

Input : integer $e \geq 0$; finite set of points $\mathcal{P} \subset \mathbb{R}^e$.

Output: minimal description of $\text{conv}(\mathcal{P})$ in terms of equations (for the affine hull of \mathcal{P}) and inequalities (one for each facet of $\text{conv}(\mathcal{P})$)

It is known, cf. Avis, Bremner & Seidel [3], Fukuda [10, Node 21], and Kaibel & Pfetsch [17, Problems 1–3], that FACETENUMERATION has a polynomial reduction to the polytope verification problem:

POLYTOPEVERIFICATION($e, \mathcal{P}, \mathcal{H}$):

Input : integer $e \geq 0$; finite set of points $\mathcal{P} \subset \mathbb{R}^e$; finite set \mathcal{H} of closed halfspaces in \mathbb{R}^e

Output: answer **yes/no** to the question whether $\text{conv}(\mathcal{P}) = \bigcap \mathcal{H}$

Freund and Orlin could show that a related problem, to decide whether $\bigcap \mathcal{H} \subseteq \text{conv}(\mathcal{P})$, is co-NP-complete [9].

3 PolytopeVerification via CompletenessG

Assuming that an LP-type oracle is available, the POLYTOPEVERIFICATION problem is polynomially equivalent to the following *geometric polytope completeness problem*:

COMPLETENESSG($d, \mathcal{V}, \mathcal{F}$):

Input : integer $d \geq 0$; finite set of points $\mathcal{V} \subset \mathbb{R}^d$; finite set \mathcal{F} of closed halfspaces in \mathbb{R}^d , such that

- $P := \text{conv}(\mathcal{V})$ is contained in $Q := \bigcap \mathcal{F}$
- $\dim P = \dim Q = d$
- every $v \in \mathcal{V}$ defines a vertex of Q
- every $F \in \mathcal{F}$ defines a facet of P

Output: answer **yes/no** to the question whether $P = Q$

As in the case of POLYTOPEVERIFICATION, the roles of vertices and facets are interchangeable for COMPLETENESSG.

We sketch the reduction of POLYTOPEVERIFICATION to COMPLETENESSG. Given any input $(e, \mathcal{P}, \mathcal{H})$ for POLYTOPEVERIFICATION, set $P := \text{conv}(\mathcal{P})$ and $Q := \bigcap \mathcal{H}$.

Employ Gaussian elimination to determine $\dim P$. Verify whether all the inequalities in \mathcal{H} are valid for P ; if this is not the case, then $P \not\subseteq Q$, so we output **no**; otherwise $P \subseteq Q$ is established. Now extract the set \mathcal{H}' of all halfspaces from \mathcal{H} for which P lies in the bounding hyperplane, that is, all those inequalities which are tight on $\text{aff } P$. An LP-type oracle is sufficient, but also needed [14], to check whether $\bigcap \mathcal{H}' = \text{aff } P$; if this is not the case, then we know that $\dim Q > \dim P$, so we can output **no**. Otherwise we proceed by restricting the input to $\text{aff } P$, that is, we deal with the situation where P is full-dimensional.

Now remove from \mathcal{H} all the halfspaces which do not determine facets of P ; this may be done using Gaussian elimination. (In the case $P = Q$, this removal does not change Q ; in the case $P \subset Q$, it may enlarge Q .) Similarly, we now remove from \mathcal{P} all those points which do not arise as intersections of some bounding hyperplanes of halfspaces in \mathcal{H} ; again this may be done via Gaussian elimination. (In the case of $P = Q$, this removal does not change P ; in the case $P \subset Q$, we may lose vertices of P , thus making P smaller.)

Now we have prepared our input for **COMPLETENESSG**. Indeed, the first two conditions on the input are satisfied, the other two are easily checked: If one of them fails, then output the answer **no**. \square

4 CompletenessG via CompletenessC

The *incidence matrix* of a polytope P with vertex set \mathcal{V} and facet set \mathcal{F} is defined to be the matrix

$$I_P := (i_{Fv})_{F \in \mathcal{F}, v \in \mathcal{V}} \in \{0, 1\}^{\mathcal{F} \times \mathcal{V}},$$

where $i_{Fv} = 1$ if vertex v lies on the facet F (that is, if $v \in F$), and $i_{Fv} = 0$ means that $v \notin F$. This matrix is well-defined up to permutation of rows and of columns, which corresponds to reordering \mathcal{V} and \mathcal{F} . A *minor* of a matrix will refer to any submatrix obtained by possibly removing rows and/or columns. A minor J of the incidence matrix I_P is *complete* if $J = I_P$. Thus we arrive at the *combinatorial polytope completeness problem*:

COMPLETENESSC(d, J):

Input : integer $d \geq 0$; incidence matrix minor J of a d -polytope

Output: answer **yes/no** to the question whether J is complete

It is not obvious that this problem is well defined. However, from Theorem 5.1 below it follows that there are no two d -polytopes P and P' such that a 0/1-matrix J is both a complete incidence matrix for P and an incomplete minor of an incidence matrix for P' . (See also the related discussion in [16].) It is clear that **COMPLETENESSG** has a polynomial reduction to **COMPLETENESSC**.

It is essential to have the dimension among the input parameters of **COMPLETENESSC**. This is demonstrated by the following example [23, p. 71]:

$$J_{\text{KM}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

We can identify $\mathcal{V} = \{1, 2, \dots, 8\}$ and $\mathcal{F} = \{1234, 1278, 1458, 2367, 3456, 5678\}$ with the sets of vertices and facets, respectively, of a 3-dimensional cube (in a suitable “Klee-Minty” vertex numbering; see Figure 1(b) below). Consequently, $\text{COMPLETENESSC}(3, J_{\text{KM}}) = \mathbf{yes}$. But we can also identify \mathcal{V} with the vertices of a cyclic 4-polytope $C_4(8)$. Then each element in \mathcal{F} corresponds to a facet of $C_4(8)$, according to Gale’s evenness criterion. Hence $\text{COMPLETENESSC}(4, J_{\text{KM}}) = \mathbf{no}$, since $C_4(8)$ has 20 facets.

A more generic class of examples for which the dimension information is needed arises from the prism construction: Let P be an arbitrary d -polytope and $P' = P \times [0, 1]$ the prism over P . The facets of P' are $P \times \{0\}$, $P \times \{1\}$, and the products of facets of P with the interval $[0, 1]$. Call the latter facets of P' *vertical*, and let J_P be an incidence matrix of P . We have $\text{COMPLETENESSC}(d, J_P) = \mathbf{yes}$. On the other hand J_P is also a minor of an incidence matrix of P' , which corresponds to the vertical facets and, say, the vertices in the bottom facet $P \times \{0\}$. Therefore, $\text{COMPLETENESSC}(d + 1, J_P) = \mathbf{no}$.

5 CompletenessC via simplicial homology

We will point out that COMPLETENESSC has a topological core. The reader is referred to Björner [5] for a survey of topological combinatorics tools, and to Munkres [21] for a presentation of simplicial homology. In the following we will use reduced simplicial homology with coefficients in \mathbb{Z}_2 . One could use any other commutative coefficient ring with unit, but \mathbb{Z}_2 is the natural choice in terms of efficiency and simplicity. We choose non-reduced homology to simplify notation for the trivial case $d = 1$.

Let $J \in \{0, 1\}^{\mathcal{F} \times \mathcal{V}}$ be an incidence matrix minor of some polytope P with vertex set $\mathcal{V}' \supseteq \mathcal{V}$ and facet set $\mathcal{F}' \supseteq \mathcal{F}$. Thus the columns of J are in bijection with a (partial) vertex set \mathcal{V} of P . Each row of J is the characteristic vector of a subset of rows, i.e., of a subset of \mathcal{V} . Thus in the following we interpret J as a combinatorial encoding of a system \mathcal{F} of (not necessarily distinct) subsets of \mathcal{V} , and with slight abuse of notation we write $\mathcal{F} \subseteq 2^{\mathcal{V}}$. The *crosscut complex* of J is the simplicial complex

$$\Gamma(J) := \left(\mathcal{V}, \bigcup \{2^F : F \in \mathcal{F}\} \right),$$

the simplicial complex of all sets of vertices that are contained in *some* facet in \mathcal{F} .

Theorem 5.1. *The incidence matrix minor $J \in \{0, 1\}^{\mathcal{F} \times \mathcal{V}}$ of a d -polytope is complete if and only if $\tilde{H}_{d-1}(\Gamma(J); \mathbb{Z}_2) \neq 0$.*

Proof. The set

$$\Pi(P, J) := \bigcup_{F \in \mathcal{F}} \text{conv}\{v \in \mathcal{V} : v \in F\} \subseteq \partial P$$

is a compact subset of the boundary of P : For every “given” facet F of P , it contains the convex hull of all “given” vertices. Thus $\Pi(P, J)$ is a polyhedral complex, called a *partial polytope*, covered by its convex (and hence contractible) cells $\text{conv}\{v \in \mathcal{V} : v \in F\}$. According to the nerve theorem [5], the crosscut complex $\Gamma(J)$ has the same homotopy type as the set $\Pi(P, J)$. In particular, the homology of the set $\Pi(P, J)$ and of the crosscut complex coincide. For an example of the crosscut complex of a partial polytope see Figure 1(a).

In the **yes** case, if the sets of vertices and facets both are complete, $\Pi(P, J)$ is the complete boundary of P , homeomorphic to S^{d-1} , so we have $\tilde{H}_{d-1}(\Gamma(\mathcal{F}); \mathbb{Z}_2) \cong \mathbb{Z}_2$.

In the **no** case, if the vertex or the facet list is incomplete, then $\Pi(P, J)$ is a proper subset of ∂P , which is a subcomplex of a suitable triangulation of ∂P , so it cannot have $(d-1)$ -dimensional homology. \square

The complexity status of the problem to compute the rank of an arbitrary homology group, or even to decide whether a certain homology group vanishes, seems to be open; see Kaibel & Pfetsch [17, Problem 33]. Thus currently our best option is based on explicitly computing simplicial homology via boundary matrices, as in Algorithm A.

Algorithm A: COMPLETENESSVIAHOMOLOGY(d, J)

Input : integer $d \geq 0$; an incidence matrix minor J of a d -polytope

Output: answer **yes/no** to the question whether J is complete

Generate \mathbb{Z}_2 -boundary matrices ∂_d and ∂_{d-1} for $\Gamma(J)$

if $\dim_{\mathbb{Z}_2} \ker \partial_{d-1} > \text{rank}_{\mathbb{Z}_2} \partial_d$ **then**

 | **return yes**

else

 | **return no**

To estimate the costs of this computation, suppose that $n = |\mathcal{V}|$, $m = |\mathcal{F}|$, and that the maximum cardinality of any facet equals s . Thus $J \in \{0, 1\}^{m \times n}$, and every row of J contains at most s ones. Then the size of the relevant boundary matrices is bounded from above by $\binom{s}{d+1}m \times \binom{s}{d}m$ and $\binom{s}{d}m \times \binom{s}{d-1}m$, respectively. We use Gaussian elimination over \mathbb{Z}_2 to compute the rank and the corank, respectively.

Corollary 5.2. *The algorithm COMPLETENESSVIAHOMOLOGY(d, J) has a polynomial running time if s is bounded by $d + c$, for an absolute constant $c \geq 0$.*

The latter case is, in fact, interesting: A d -polytope is *simplicial* if each proper face is a simplex or, equivalently, each facet contains exactly d vertices. We infer that the running time of COMPLETENESSVIAHOMOLOGY for simplicial polytopes is bounded by $O(dm^3)$.

It has been observed by Bremner, Fukuda & Marzetta [7] that FACETENUMERATION for a polytope P is polynomially equivalent to FACETENUMERATION for the dual polytope P^* . Using our techniques, a similar result can be obtained directly. If I is an incidence matrix for P , then the transposed matrix I^{tr} is an incidence matrix for P^* . Any minor J of I is complete if and only if its transpose is a complete minor of I^{tr} . This leads to the following modification of our algorithm. While s was defined above as the maximal row size of the input incidence matrix minor, define

$$s' := \min\{\text{maximal row size, maximal column size}\}.$$

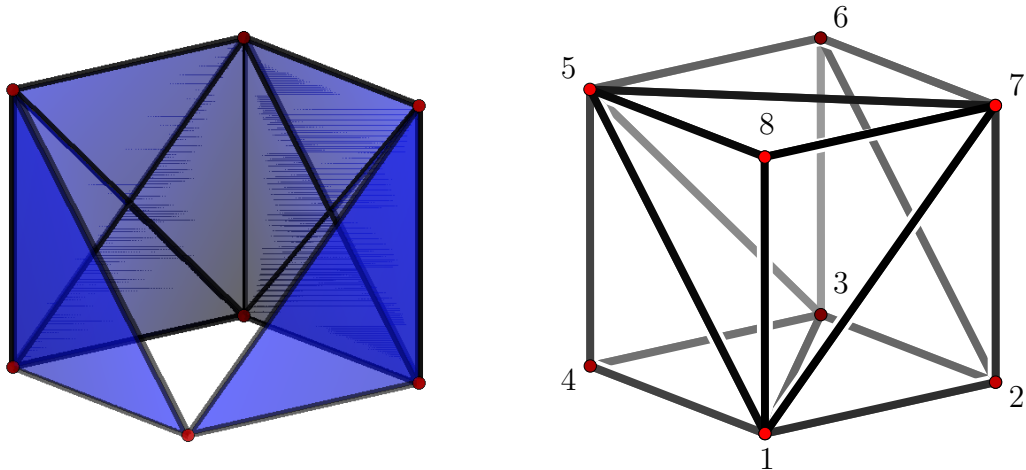
Thus we modify our algorithm: It should first compare the sizes of the primal and the dual problem, and then perform the (reduced) homology computation for the smaller problem. The modified algorithm COMPLETENESSVIAHOMOLOGY(d, J) has polynomial running

time if s' is bounded by “ d plus a constant.” In particular, this yields an $O(d(n+m)^3)$ -algorithm for the COMPLETENESSC problem specialized to polytopes which are simplicial or *simple*, that is, dual to a simplicial polytope.

We note, however, that these running times are neither optimal nor the best available: The reverse search algorithm of Avis and Fukuda [4] computes the convex hull (and thereby solves COMPLETENESSG) of a simplicial polytope in $O(dnm)$ steps.

6 A Certificate for Incompleteness

Let P be a d -polytope with ordered vertex set $\mathcal{V}' = \{v_1, \dots, v_n\}$ and facet set \mathcal{F}' . Inductively, define a sequence $\Delta_0, \dots, \Delta_n$ of polytopal subdivisions of the boundary complex ∂P : Set $\Delta_0 := \partial P$. In order to obtain Δ_k replace each facet F of Δ_{k-1} which contains v_k by the set of cones with apex v_k over those facets of F which do not contain v_k . The final subdivision is a triangulation $\Delta(P) := \Delta_m$ of ∂P , the *pulling triangulation* [18] with respect to the chosen ordering of \mathcal{V} . For an example of a pulling triangulation see Figure 1(b).



(a) The (3-dimensional) crosscut complex of some partial 3-cube C . The two quadrangle faces of C yield tetrahedra in $\Gamma(C)$, which are displayed almost flat.

(b) The pulling triangulation of the boundary of a 3-cube with respect to a “Klee-Minty” vertex ordering. The facet $\{1, 7, 8\}$ of the triangulation corresponds to the flag $\{8\} \subset \{7, 8\} \subset \{1, 2, 7, 8\}$ of the cube.

Figure 1: Crosscut complex and pulling triangulation.

The pulling triangulation of ∂P has several nice properties (not shared, for example, by the “placing triangulation”) which may be exploited for our purposes. First, its combinatorics is determined by the combinatorics of P ; see below. Furthermore, if we use a linear ordering of the vertex set \mathcal{V}' in which the vertices in \mathcal{V} come first, then the corresponding pulling triangulation of the boundary of P contains $\Pi(P, J)$ as a subcomplex.

Let us now identify the vertex set \mathcal{V}' with the set $[n] = \{1, \dots, n\}$ and each facet $F \in \mathcal{F}'$ with the subset of $[n]$ that corresponds to the vertices contained in F . Thus any

triangulation of ∂P is encoded by a collection of d -subsets of $[n]$, that is, to a subset of $\binom{[n]}{d}$. We write $\{v_1, \dots, v_d\}_<$ for a d -subset of $[n]$ with $v_1 < v_2 < \dots < v_d$.

Lemma 6.1. *Let P be a d -polytope whose vertex set is labeled by $[n]$.*

Then a set $\{v_1, \dots, v_d\}_< \in \binom{[n]}{d}$ corresponds to a facet of the pulling triangulation of ∂P (with respect to the chosen vertex labeling) if and only if there is a complete flag of faces

$$\emptyset \subset G_0 \subset G_1 \subset \dots \subset G_{d-1} \subset P,$$

such that v_i is the smallest vertex in G_{d-i} for $1 \leq i \leq d$, that is, if there are facets F_1, \dots, F_d of P such that

$$v_i = \min(F_1 \cap \dots \cap F_i)$$

for $1 \leq i \leq d$.

Proof. Every pulling facet $\{v_1, \dots, v_d\}_<$ lies in a facet $F_1 = G_{d-1}$ of P , with $v_1 = \min G_{d-1}$. It is a cone with apex v_1 and base $G_{d-2} \subset G_{d-1}$. The existence of the rest of the maximal flag $(G_i)_{0 \leq i < d}$ follows recursively. Given the flag, the existence of the facets F_1, \dots, F_d follows [23, Lect. 2]. Given a complete flag, the corresponding sequence of facets F_i is uniquely determined if P is simple, but not in general. \square

If we have an arbitrary incidence matrix minor J of a d -polytope P , then we can read the combinatorial characterization of the pulling triangulation from Lemma 6.1 as the definition of a complex that coincides with the pulling triangulation of ∂P in case J is complete, but is well-defined in general:

Definition 6.2. Given an integer $d > 0$ and a 0/1-matrix $J \in \{0, 1\}^{m \times n}$, which we interpret as the incidence matrix of a set system $\mathcal{F} \subseteq 2^{[n]}$, the *pulling complex* of d and J is

$$\Delta(d, J) := \left\{ \{v_1, \dots, v_d\}_< \in \binom{[n]}{d} : \begin{array}{l} \text{there are } \bar{F}_1, \dots, \bar{F}_d \in \mathcal{F} \text{ such that} \\ v_i = \min(\bar{F}_1 \cap \dots \cap \bar{F}_i) \text{ for } 1 \leq i \leq d \end{array} \right\}.$$

Lemma 6.3. *Let P be a d -dimensional polytope with vertex set \mathcal{V}' and facet set \mathcal{F}' , and let J be a incidence matrix minor corresponding to subsets $\mathcal{V} \subseteq \mathcal{V}'$ and $\mathcal{F} \subseteq \mathcal{F}'$. Let $\bar{P} \subseteq P$ be the convex hull of the vertices in \mathcal{V} . Fix a linear ordering on the vertex set \mathcal{V}' such that the vertices in \mathcal{V} come first.*

Then the simplicial complex $\Delta(d, J)$ is a subcomplex of $\Delta(P)$ as well as of $\Delta(\bar{P})$. In particular, $\Delta(d, J)$ is a proper subcomplex of $\Delta(P)$, unless the minor J is complete, $J = I_P$. In the incomplete case $\Delta(d, J)$ may even be empty.

Proof. Let $\{v_1, \dots, v_d\}_< \in \Delta(d, J)$, then there are $\bar{F}_1, \dots, \bar{F}_d \in \mathcal{F}$ such that $v_i = \min(\bar{F}_1 \cap \dots \cap \bar{F}_i)$. Now since J is an incidence matrix minor of P , there are facets $F_i \supseteq \bar{F}_i$ of P , and by the assumption on the vertex ordering the vertices in \bar{F}_i come first, so $\min(\bar{F}_1 \cap \dots \cap \bar{F}_i) = \min(F_1 \cap \dots \cap F_i)$, which yields $\{v_1, \dots, v_d\}_< \in \Delta(P)$.

Now $\bar{P} = \text{conv}(\mathcal{V})$, and the $\bar{F}_i = F_i \cap \mathcal{V}$ are vertex sets of faces (not necessarily facets) of \bar{P} . If the vertices $v_i = \min(\bar{F}_1 \cap \dots \cap \bar{F}_i)$ are distinct, then the faces $\bar{F}_1 \cap \dots \cap \bar{F}_i$ form a complete flag in the face lattice of \bar{P} , and thus $\{v_1, \dots, v_d\}_< \in \Delta(\bar{P})$, by Lemma 6.1. \square

In particular, $\Delta(d, J)$ triangulates a subset of the complex $\Pi(P, J)$ that appears in the proof of Theorem 5.1.

Now we present a polynomially-checkable certificate for the case that J is incomplete. Note, however, that this result does *not* prove that **COMPLETENESSC** is in co-NP: We are not able to check (in polynomial time) whether the input is valid, that is, whether J is actually an incidence matrix minor of some d -polytope.

Theorem 6.4. *Any no instance of the problem **COMPLETENESSC**(d, J) has a certificate that can be verified in polynomial time.*

Proof. The minor J is incomplete if and only if the pulling complex $\Delta(d, J)$ is not a complete triangulation of a d -polytope boundary. Two cases arise. The first one is if $\Delta(d, J) = \emptyset$, in which case Algorithm B described below will certify in polynomial time that J is not complete.

The second case is if $\Delta(d, J)$ is non-empty but incomplete. In this case (since the dual graph of the pulling triangulation $\Delta(P)$ is connected) there is a facet $\{v_1, \dots, v_d\} \in \Delta(d, J)$ together with an index i such that there is no second facet of $\Delta(d, J)$ that contains $\{v_1, \dots, v_d\} \setminus \{v_i\}$. In this situation our certificate is the set $\{v_1, \dots, v_d\} \setminus \{v_i\}$. Calling **ISPULLINGFACET** for every d -subset of $[n]$ which contains the certificate, this certificate can be verified in polynomial time, since there are $n - d + 1$ of these subsets. \square

Now we proceed by describing the two subroutines needed for Theorem 6.4. The first one is Algorithm B: Given an incidence matrix minor J it either finds a facet of $\Delta(d, J)$ in polynomial time or it detects that J is incomplete. The correctness follows from Lemma 6.1. Our specific formulation of the algorithm produces a pulling triangulation facet which does not contain 1: This restriction does not hurt, since $\Delta(d, J)$ must contain such a facet if J is complete.

Algorithm B: **FINDPULLINGFACET**(d, J)

Input : incidence matrix minor $J \in \{0, 1\}^{m \times n}$ of a d -polytope; d -tuple $\{v_1, \dots, v_d\}_< \in \binom{[n]}{d}$

Output: a facet $\{v_1, \dots, v_d\} \in \Delta(d, J)$, or **incomplete**

$S \leftarrow [n]$

for $i \leftarrow 1$ **to** d **do**

$F_i \leftarrow$ any $F \in \mathcal{F}$ such that $\min S \notin F$, $F \cap S \neq \emptyset$, and $|F \cap S|$ is maximal
if no such facet exists **then**
└ **return incomplete**
 $S \leftarrow S \cap F_i$
 $v_i \leftarrow \min S$

return $\{v_1, \dots, v_d\}_<$

Our second subroutine, Algorithm C, checks whether a given set of d vertices is a facet of the pulling complex $\Delta(d, J)$ or not. Its correctness again follows from the characterization in Lemma 6.1. Its running time is bounded by $O(d(n + m))$.

Algorithm C: $\text{ISPULLINGFACET}(d, J, \{v_1, \dots, v_d\}_{<})$

Input : (d, J) as above

Output: answer **yes/no** to the question whether $\{v_1, \dots, v_d\} \in \Delta(d, J)$

for $i \leftarrow d$ *downto* 1 **do**

└ compute the set \mathcal{F}_i of all facets (i.e., rows of J) that contain $\{v_i, \dots, v_d\}$

for $i \leftarrow 1$ *to* d **do**

└ $F_i \leftarrow$ any $F \in \mathcal{F}_i$ with $v_i = \min(F_1 \cap \dots \cap F_{i-1} \cap F)$

└ **if** *no such F exists* **then**

└└ **return no**

return yes

We close our discussion with a pointer to a specific special case: It would be interesting to know whether $\text{COMPLETENESS}(d, J)$ has a polynomial time solution for the very special case where J has all columns and lacks at most one row.

Acknowledgements

We are grateful to Volker Kaibel, Marc E. Pfetsch and Mark de Longueville for helpful comments. Moreover, the first author is indebted to Günter Rote for an enlightening discussion on the subject.

References

- [1] D. Avis: *lrs* — a C implementation of the reverse search vertex enumeration algorithm, version 4.1. <http://cgm.cs.mcgill.ca/~avis/C/lrs.html>
- [2] ———: *lrs: A revised implementation of the reverse search vertex enumeration algorithm*, in: “Polytopes — Combinatorics and Computation” (G. Kalai & G. M. Ziegler, eds.), *DMV-Seminars* Vol. 29, Birkhäuser Basel 2000, pp. 177–198.
- [3] D. Avis, D. Bremner & R. Seidel: *How good are convex hull algorithms?*, *Comput. Geom.* **7** (1997), 265–301.
- [4] D. Avis & K. Fukuda: *A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra*, *Discrete Comput. Geom.* **8** (1992), 295–313.
- [5] A. Björner: *Topological methods*, in: “Handbook of Combinatorics” (R. L. Graham, M. Grötschel & L. Lovász, eds.), Vol. II, North-Holland, Amsterdam 1995, pp. 1819–1872.
- [6] D. Bremner: *Incremental convex hull algorithms are not output sensitive*, *Discrete Comput. Geom.* **21** (1999), 57–68.
- [7] D. Bremner, K. Fukuda & A. Marzetta: *Primal-dual methods for vertex and facet enumeration*, *Discrete Comput. Geom.* **20** (1998), 333–357.
- [8] B. Chazelle: *An optimal convex hull algorithm in any fixed dimension*, *Discrete Comput. Geom.* **10** (1993), 377–409.
- [9] R. M. Freund & J. B. Orlin: *On the complexity of four polyhedral set containment problems*, *Math. Programming* **33** (1985), 139–145.

- [10] K. Fukuda: *Frequently asked questions in polyhedral computation*, Oct. 2000.
<http://www.cs.mcgill.ca/~fukuda/soft/polyfaq/polyfaq.html>
- [11] ———: CDD — A C-implementation of the double description method; cddlib, version 0.92b. http://www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html
- [12] E. Gawrilow & M. Joswig: polymake, version 1.5.1: a software package for analyzing convex polytopes, 1997–2003. <http://www.math.tu-berlin.de/polymake>
- [13] ———: *polymake: an approach to modular software design in computational geometry*, in: Proc. 17th Annual Symp. Computational Geometry, ACM, 2001, June 3-5, 2001, Medford, MA, pp. 222–231.
- [14] M. Grötschel, L. Lovász & A. Schrijver: *Geometric Algorithms and Combinatorial Optimization, Algorithms and Combinatorics*, Vol. 2, Springer-Verlag, Berlin Heidelberg, 1988; second edition 1994.
- [15] M. Joswig: *Beneath-and-beyond revisited*, to appear in “Algebra, Geometry, and Software Systems” (M. Joswig & N. Takayama, eds.), Springer-Verlag, Heidelberg 2003, pp. 1–21.
- [16] M. Joswig, V. Kaibel, M. E. Pfetsch & G. M. Ziegler: *Vertex-facet incidences of unbounded polyhedra*, Adv. Geometry **1** (2001), 23–36.
- [17] V. Kaibel & M. E. Pfetsch: *Some algorithmic problems in polytope theory*, in: “Algebra, Geometry, and Software Systems” (M. Joswig & N. Takayama, eds.), Springer-Verlag, Heidelberg 2003, pp. 23–48.
- [18] C. W. Lee: *Subdivisions and triangulations of polytopes*, in: “Handbook of Discrete and Computational Geometry” (J. E. Goodman & J. O’Rourke, eds.), CRC Press, Boca Raton 1997, pp. 271–290.
- [19] N. McCarthy, D. Ogilvie, I. Spitkovsky & N. Zobin: *Birkhoff’s theorem and convex hulls of Coxeter groups*, Linear Algebra Appl., **347** (2002), 219–231.
- [20] P. McMullen: *The maximum numbers of faces of a convex polytope*, Mathematika **17** (1970), 179–184.
- [21] J. R. Munkres: *Elements of Algebraic Topology*, Addison-Wesley, Menlo Park, CA 1984.
- [22] R. Seidel: *Small-dimensional linear programming and convex hulls made easy*, Discrete Comput. Geom. **6** (1991), 423–434.
- [23] G. M. Ziegler: *Lectures on Polytopes*, *Graduate Texts in Math.*, Vol. 152, Springer-Verlag, New York 1995; revised 1998.