Point Containment in the Integer Hull of a Polyhedron

Ernst Althaus† Friedrich Eisenbrand† Stefan Funke† Kurt Mehlhorn†

Abstract

We show that the point containment problem in the integer hull of a polyhedron, which is defined by m inequalities, with coefficients of at most φ bits can be solved in time $O(m+\varphi)$ in the two-dimensional case and in expected time $O(m+\varphi^2 \log m)$ in any fixed dimension. This improves on the algorithm which is based on the equivalence of separation and optimization and on a direct algorithm (SODA 97) for the two-dimensional case.

1 Introduction

We are interested in the *point containment problem in integer hulls of polyhedra*: Given a point $x^* \in \mathbb{Q}^d$ and a set of rational constraints $Ax \leq b$, $A \in \mathbb{Q}^{md}$, $b \in \mathbb{Q}^m$, determine whether x^* belongs to the convex hull of the integral points satisfying the constraints. Moreover, *certify* your answer by providing a simplex containing x^* which is spanned by feasible integer points in the "yes" case, or by providing a halfspace h containing x^* such that $h \cap P$ is integer infeasible in the "no" case. We use $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$ to denote the polyhedron defined by our set of constraints and P_I to denote the convex hull of the integral points in P; P_I is frequently called the *integer hull* of P.

Let m be the number of constraints, d the dimension of ambient space, and assume that each constraint and x^* has binary encoding length $O(\varphi)$. We show:

THEOREM 1.1. *For $d=2$, the point containment problem in integer hulls of polygons can be solved in time $O(m+\varphi)$. For $d \geq 3$ and d fixed, the point containment problem in integer hulls of polyhedra can be solved in expected time $O(m+\varphi^2 \log m)$.*

We will make frequent use of the fact that integer programming can be done in expected time $O(m+\varphi \log m)$ in any fixed dimension [3] and in time $O(m+\varphi)$ in the two-dimensional case [4]. Also the integer hull of a polygon (vertices in clockwise order) can be computed in time $O(m\,\varphi)$ in two dimensions [6]. We also assume without loss of generality [11] that P is bounded and that P_I is full-dimensional.

[1]Partially supported by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT)

[2]Max-Planck-Institute for Computer Science, Saarbrücken, Germany

2 Related work

In two dimensions ($d=2$), McCormick, Smallwood and Spieksma [9, 8] developed an algorithm, which runs in time $O(m\,\varphi+\varphi^2)$. Using the equivalence of optimization and separation [5] together with recent algorithms for integer programming [3, 4] one can solve the point containment problem with the ellipsoid method. This yields an expected running time of $O(m\,\varphi+\varphi^2 \log m)$ for $d \geq 3$ and a running time of $O(m\,\varphi+\varphi^2)$ for $d=2$. These algorithms are certifying in our sense.

McCormick et al. [9, 8] reduce a multiprocessor machine scheduling problem to the two-dimensional point-containment problem, where containment has to be certified with a unimodular triangle, i.e., with a triangle that does not contain any integer points besides its vertices. Given any feasible integer triangle T which contains x^*, one can construct a unimodular triangle T_u which contains x^* as follows.

Compute the integer hulls L and R of the two polygons $T \cap (x(1) \leq \lfloor x^*(1) \rfloor)$ and $T \cap (x(1) \geq \lceil x^*(1) \rceil)$. The closure of the set $T \setminus (L \cup R)$ is a (not necessarily convex) polygon B which contains x^*. This polygon can be computed in time $O(\varphi)$ and has $O(\varphi)$ vertices. Now triangulate B and determine the triangle T' containing x^*. This costs again $O(\varphi)$ [1]. The interior of T' does not contain an integer point and only one edge e of T', the edge stemming from an edge of B, might contain other integer points. Consider the intersection y^* of the ray $\overrightarrow{v, x^*}$, where v is the opposite vertex of e, with this edge e. The two nearest integer points of y^* on e, together with v form a certifying unimodular triangle. These two nearest points can be found by solving a one-dimensional integer program, or directly, with one extended gcd-computation.

3 An algorithm for $d=2$

For a given point $x^* \in \mathbb{Q}^2$, the following simple algorithm solves the point containment problem in the integer hull of a polyhedron $P \subseteq \mathbb{Q}^2$ in time $O(m+\varphi)$, see Figure 1.

1. Find an integer point $u \in P$. If $x^* \in P_I$,

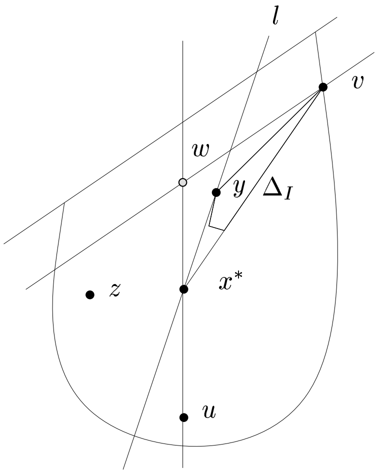

Figure 1: The case $d = 2$.

then x^* is contained in an integral triangle with vertex u.

2. Determine the constraint $a^T x \leq \beta$, which defines the facet of P which is hit by the ray $\overrightarrow{u, x^*}$ (ties are broken arbitrarily).

3. Find the optimal integer point v in P w.r.t. to the objective function $\max a^T x$; let β^* be the optimal objective function value.

4. Let w be the intersection of the line $a^T x = \beta^*$ with $\overrightarrow{u, x^*}$. If $a^T x^* > \beta^*$, then x^* is not contained in P_I and we have found a certifying hyperplane, otherwise consider the triangle $\Delta = \text{conv}(x^*, v, w)$ and compute its integer hull Δ_I. Note that Δ is contained in P.

5. Compute the line l which intersects x^* and is tangential to Δ_I such that u and Δ_I lie on the same side of l. Let y be the first vertex of Δ_I which lies on l, starting from x^*.

6. Perform an integer feasibility test for P intersected with the halfspace h defined by the closure of the side of l which does not contain Δ_I. Any integer point z in the resulting polygon must be opposite of y on the line through u and x^*. Thus $\text{conv}(u, y, z)$ is a certifying triangle. If no integer point exists in this polygon, then x^* is not contained in P_I. A certifying hyperplane can then be determined by slightly rotating the line l around y. This rotation can be determined by similar means with which we determined y in Step 5.

For the running time, observe that we have to solve a constant number of optimization problems ($O(m + \varphi)$) and perform one integer hull computation for a constant number of constraints ($O(\varphi)$); finding the tangent line can also be done in $O(\varphi)$ as Δ_I has $O(\varphi)$ vertices. This yields a final running time of $O(m + \varphi)$.

4 Point containment in arbitrary fixed dimension

Let us now consider the case $d \geq 3$. We assume that the integer hull of the polyhedron P is not empty, which can be tested with a call to an optimization oracle. Further we assume without loss of generality that P is bounded. We first describe a simple procedure to solve the problem, which has a worse running time than the desired $\mathrm{O}(m + \varphi^2 \log m)$. Later, we plug this procedure into the framework of LP-type problems [7] to obtain the claimed running time.

Given a polytope $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$, the following procedure computes a unique simplex Σ with vertices in P_I and $x^* \in \Sigma$ if and only if $x^* \in P_I$.

ContainingSimplex(P, x^*)

1. Compute the lexicographically smallest integer point u of P_I.

2. Determine a face-defining inequality $a^T x \leq \beta$ of the minimal face of P_I that is hit by the ray $\overrightarrow{u, x^*}$ and let w be the intersection point. If $u = x^*$ or of this ray does not hit the boundary of P_I in a point different from u, then return the simplex $\Sigma = \{u\}$.

3. Otherwise, recursively determine the simplex Σ' containing w in the integer hull of $P' = P \cap (a^T x = \beta)$.

4. Return the simplex spanned by u and Σ'.

The so constructed simplex is unique and denoted by $\Sigma(x^*, Ax \leq b)$.

Before we begin with an analysis of this approach, let us define the *height* $\nu(x^*, Ax \leq b)$ of x^* and $Ax \leq b$, which is a tuple. If Σ consists of u alone, then $\nu(x^*, Ax \leq b) = (u)$. Otherwise, let h be the distance from u to w. The height is then recursevly defined as the tuple $\nu(x^*, Ax \leq b) = (u, -h, \nu(x^*, Ax \leq b \wedge a^T x = \beta)$. In the following, we assume that the height $\nu(x^*, Ax \leq b)$ is a $2d + 1$-tuple by appending ∞-components the tuple defined above. We define an order on sets of constraints as $(Ax \leq b) \prec (Bx \leq c)$ if $\nu(x^*, Ax \leq b) \leq_{\text{lex}} \nu(x^*, Bx \leq c)$. Notice that $\nu(x^*, Ax \leq b) = \nu(x^*, Bx \leq c)$ if and only if the simplices $\Sigma(x^*, Ax \leq b)$ and $\Sigma(x^*, Bx \leq c)$ are equal.

Step 1 of the above procedure takes an expected number of $\mathrm{O}(m + \varphi \log m)$ steps, whereas step 2 can be solved in expected time $O(m\varphi + \varphi^2 \log m)$ as follows:

Let $(x_i)_{i \in I}$ be the set of vertices of P_I. The intersection point w of the ray starting at u through x^* with the boundary of P_I is the last point of the ray that is a convex combination of points in $(x_i)_{i \in I}$. Thus the following linear program finds the intersection point w.

$$\begin{array}{lrcl} \max & \alpha & & \\ s.t. & u + \alpha(x^* - u) & = & \sum_{i \in I} \lambda_i x_i \\ & \sum_{i \in I} \lambda_i & = & 1 \\ & \lambda & \geq & 0 \end{array}$$

The dual of this linear program has the following form:

$$\begin{array}{lrcl} \min & y^T u + z & & \\ s.t. & y^T(x^* - u) & = & 1 \\ & -y^T x_i + z & \geq & 0 \end{array}$$

It has $d+1$ variables. The separation problem for the dual is an optimization over the integer hull of P. Thus the separation problem can be solved in time $O(m + \varphi \log m)$ and hence the linear program can be solved via the ellipsoid method [?] in time $O(m\,\varphi + \varphi^2 \log m)$.

It remains to show, how to compute the face-defining inequality $a^T x \leq \beta$ of the minimal face of P_I containing w. This is done as follows, see also [5, p. 183, theorem 6.5.8]. Let Q be the polytope $Q = \{(z^T, \mu)^T \in \mathbb{R}^{d+1} \mid z^T x \leq \mu \text{ for all } x \in P_I,\, z^T w = \mu\}$, which is the polytope of valid inequalities for P_I which are tight at w. The separation problem for this polytope is again an optimization problem over the integer hull of P. Thus the optimization problem over Q can be solved in time $O(m\,\varphi + \varphi^2 \log m)$. Within this time-bound one can find a point $(a^T, \beta)^T$ in the relative interior of Q. The inequality $a^T x \leq \beta$ is the face-defining inequality we are looking for.

Therefore the overall running time of this procedure on a polytope defined by m constraints in d dimensions is $O(m\varphi + \varphi^2 \log m)$ for fixed d. So the running time is dominated by Step 2. In the following we apply the machinery of LP-type problems such that this step has to be performed $O(\log m)$ times on subproblems of constant size.

We begin with the following observation.

LEMMA 4.1. *Let $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$ be a polytope in fixed dimension d with nonempty integer hull. Then there exists a subset $A'x \leq b'$ of the constraints $Ax \leq b$ such that the simplices $\Sigma(x^*, Ax \leq b)$ and $\Sigma(x^*, A'x \leq b')$ are equal.*

Proof. The lexicographically minimal vertex of P_I can be found by solving an integer optimization problem over P_I (remember that P is a polytope). It follows from a lemma by Scarf [10], see also [11, p. 234] that there exists a constant-size subset of the constraints $Ax \leq b$ such that the bottom-vertex w.r.t. the integer hull of this subset and the bottom-vertex of P_I coincide. Similarly, we can find a constant-size subset of $Ax \leq b$ such that the face defining inequality $a^T x \leq \beta$ of the construction above, is also face-defining for the integer-hull of this subset. The assertion then follows by induction.

We now model the search for a constant size subset $A'x \leq b'$ of $Ax \leq b$, such that $\Sigma(x^*, Ax \leq b) = \Sigma(x^*, A'x \leq b')$ as an *LP-type problem.* Such an LP-type problem [7] is specified by a pair (H, ω), where H is a finite set, whose elements are called *constraints* and $\omega : 2^H \mapsto \mathcal{W}$ is a function with values in a linearly ordered set $(\mathcal{W}, \leq)$ satisfying a certain set of axioms (see below). The goal is to compute a minimal subset B_H of H with the same value as H. In our setting the set H consists of the constraints defining P and the order is the order on sets of constraints defined above.

The following axioms are satisfied.

Axiom 1.*(Monotonicity)* For any F, G with $F \subseteq G \subseteq H$, we have $\omega(F) \leq \omega(G)$.

The height is invariant under different representations of the polytope. To show the validity of this axiom, we therefore only use the fact that the (integral) polytopes $P(F)_I$ and $P(G)_I$ satisfy $P(F)_I \supseteq P(G)_I$.

The bottom vertex of $P(F)_I$ is less then or equal the bottom vertex of $P(G)_I$. Assume now that they are the same. If the second component of $\omega(G)$ is infinity, then we have $\omega(F) \leq \omega(G)$. Otherwise, this second component is a negiative number which represents minus the distance on the ray to the facet of $P(G)_I$. This distance is at most the distance to the facet of $P(F)_I$.

Assume now that the bottom-vertices and positive distances coincide. Thus the intersection points coincide. Let w denote the intersection point. Consider the face defining inequality $a^T x \leq \beta$ for the minimal face of $P(F)_I$ containing w and the corresponding face-defining inequality ${a'}^T x \leq \beta'$ for the minimal face of $P(G)_I$. Clearly $a^T x \leq \beta$ is a face-defining inequality of $P(G)_I$, whose induced face contains the minimal face of $P(G)_I$, containing w. Thus $P(F \wedge a^T x = \beta) \supseteq P(G \wedge a^T x = \beta) \supseteq P(G \wedge {a'}^T x = \beta')$. Thus $P(F \wedge a^T x = \beta)_I \supseteq P(G \wedge {a'}^T x = \beta')_I$ and the axiom follows by induction.

Axiom 2.*(Locality)* For any $F \subseteq G \subseteq H$ with $\omega(F) = \omega(G)$ and any $h \in H$, $\omega(G) < \omega(G \cup \{h\})$ implies that also $\omega(F) < \omega(F \cup \{h\})$.

If $\omega(F) = \omega(G)$ holds, then the simplices $\Sigma(x^*, F)$ and $\Sigma(x^*, G)$ coincide. If $\omega(G) < \omega(G \cup$

$\{h\})$, then h cuts off a vertex of this simplex. Consequently also $\omega(F)$ strictly increases.

A basis B_G of a subset $G \subseteq H$ is a minimal subset of G with $\omega(B_G) = \omega(G)$. The combinatorial dimension of a LP-type problem is the size of the largest basis of any $G \subseteq H$. From Lemma 4.1, we know that the combinatorial dimension of our problem is constant.

An LP-type problem of constant combinatorial dimension can be solved with $O(m)$ *violation tests* and $O(\log m)$ *basis computations* for constant size subsets of constraints, as shown in [2, 7]. A violation test for a basis B and a constraint h is the problem of determining whether $\omega(B) \neq \omega(B \cup \{h\})$. A basis computation for a set of constraints G is the task of computing a basis of G.

Lets first deal with the violation test. Let B be a basis. A constraint violates this basis, if and only if it cuts off (at least) one of the corner points of $\Sigma(x^*, B)$. Thus, we iterate over the corners of the simplex and check violation. This requires constant time.

For the basis computation, let $G \subseteq H$ be a subset of the constraints of constant size. We have to compute the simplex $\Sigma(x^*, G)$. This can be done by calling our procedure **ContainingSimplex(**G**,** x^***)**. As G has constant size, this call will cost $\mathrm{O}(\varphi^2)$ and by brute-force, we can determine which constraints are necessary to obtain the same simplex. This concludes the proof of Theorem 1.1.

References

[1] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6(5):485–524, 1991.

[2] K. L. Clarkson. Las vegas algorithms for linear and integer programming when the dimension is small. *Journal of the Association for Computing Machinery*, 42:488–499, 1995.

[3] F. Eisenbrand. Fast integer programming in fixed dimension. Technical Report MPI-I-2003-NWG2-002, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 2003. to appear in the Proceedings of ESA 2003.

[4] F. Eisenbrand and S. Laue. A faster algorithm for two-variable integer programming. manuscript, `http://www.mpi-sb.mpg.de/~eisen/2ip.ps.gz`.

[5] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[6] W. Harvey. Computing two-dimensional integer hulls. *SIAM Journal on Computing*, 28(6):2285–2299, 1999.

[7] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4-5):498–516, 1996.

[8] S. T. McCormick, S. R. Smallwood, and F. C. R. Spieksma. Polynomial algorithms for multiprocessor scheduling with a small number of job lengths. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (New Orleans, LA, 1997)*, pages 509–517, New York, 1997. ACM.

[9] S. T. McCormick, S. R. Smallwood, and F. C. R. Spieksma. A polynomial algorithm for multiprocessor scheduling with two job lengths. *Mathematics of Operations Research*, 26(1):31–49, 2001.

[10] H. E. Scarf. An observation on the structure of production sets with indivisibilities. *Proc. Nat. Acad. Sci. U.S.A.*, 74(9):3637–3641, 1977.

[11] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley, 1986.