Point Containment in the Integer Hull of a Polyhedron[1]

Ernst Althaus[2] Friedrich Eisenbrand[2] Stefan Funke[2] Kurt Mehlhorn[2]

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 in the general case 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 $Ax \leq b$, $A \in \mathbb{Q}^{m \times d}$, $b \in \mathbb{Q}^m$, 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], in particular the number of vertices of the integer hull is $O(m\,\varphi)$. We also assume without loss of generality [11] that P is bounded and that P_I is full-dimensional.

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]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

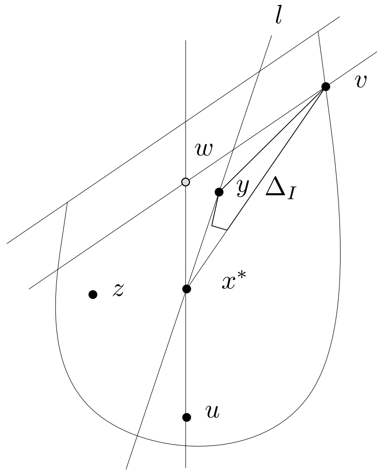

Figure 1: The case $d = 2$.

1. Find an integer point $u \in P$. If $x^* \in P_I$, 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 and that implicit upper and lower bounds on the variables are part of the constraint set $Ax \leq b$. Recall that an inequality $a^T x \leq \beta$ is called *valid* for P, if all points of P satisfy it. A subset of points $F \subseteq P$, which satisfy a valid inequality $a^T x \leq \beta$ of P with equality is called a *face* of P. The inequality $a^T x \leq \beta$ is then called the *face-defining inequality* of F.

Given a polytope P, the following procedure computes a 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 the last point in P_I on the ray $\overrightarrow{u, x^*}$ and denote it by w. Furthermore, determine a face-defining inequality $a^T x \leq \beta$ of the minimal face of P_I containing w. If $u = x^*$ or if $u = w$, then return the simplex $\Sigma = \{u\}$.

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

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

Before we begin with an analysis of this approach, let us define the *height* $\nu(x^*, P)$ of x^* and P, which is a tuple. If the simplex $\Sigma(x^*, P)$ consists of u alone, then $\nu(x^*, P) = (u)$. Otherwise, let h be the distance from u to w. The height is then recursively defined as the tuple $\nu(x^*, P) = (u, -h, \nu(w, P \cap (a^T x = \beta))$. In the following, we assume that the height $\nu(x^*, P)$ is a $2\,d+1$-tuple by appending ∞-components to right of the tuple defined above. Given x^*, we define an order $P \preceq_{x^*} Q$ if $\nu(x^*, P) \leq_{\text{lex}} \nu(x^*, Q)$ for polytopes $P, Q \subseteq \mathbb{R}^d$. Notice that $\nu(x^*, P) = \nu(x^*, Q)$ if and only if the simplices $\Sigma(x^*, P)$ and $\Sigma(x^*, Q)$ are equal. Furthermore we have the following lemma.

LEMMA 4.1. *Let $P \subseteq \mathbb{R}^d$ and $Q \subseteq \mathbb{R}^d$ be polytopes and $x^* \in \mathbb{R}^d$. If $P \subseteq Q$, then $\nu(x^*, Q) \leq_{\text{lex}} \nu(x^*, P)$ and* *thus $Q \preceq_{x^*} P$. Moreover, $\nu(x^*,Q) <_{\text{lex}} \nu(x^*,P)$ iff a vertex of $\Sigma(x^*,Q)$ is not contained in P.*

Proof. Let $\Sigma_P = \Sigma(x^*,P)$ and $\Sigma_Q = \Sigma(x^*,Q)$.

Let u_P and u_Q, denote the lexicographically smallest integer point of P and Q respectively. Since $u_P \in Q$ we have $u_Q \leq_{\text{lex}} u_P$. If they differ, we have $\nu(x^*,Q) <_{\text{lex}} \nu(x^*,P)$ and a vertex of Σ_Q is not contained in P. We now assume that $u_P = u_Q$ and denote this point by u. If $u = x^*$, $\Sigma_Q = \Sigma_P = \{u\}$ and $\nu(x^*,P) = (u,\infty,\ldots,\infty) = \nu(x^*,Q)$. So assume $u \neq x^*$ and let w_P and w_Q be the last vertices in the boundary of P_I and Q_I, respectively, on the ray $\overrightarrow{u,x^*}$ and let F_P and F_Q be the minimal faces containing these points, respectively. Since $P_I \subseteq Q_I$, w_Q does not precede w_P on $\overrightarrow{u,x^*}$. If $w_Q = u$, $\Sigma_Q = \Sigma_P = \{u\}$ and $\nu(x^*,P) = (u,\infty,\ldots,\infty) = \nu(x^*,Q)$. So assume $w_Q \neq u$. If $w_P \neq w_Q$, $w_Q \notin P_I$ and $\nu(x^*,Q) <_{\text{lex}} \nu(x^*,P)$. Also, $w_Q \in F_Q$ and hence is contained in the simplex Σ'_Q, where Σ'_Q is the simplex determined in step 3 of **ContainingSimplex(Q, x^*)**. Since $w_Q \notin P_I$, one of the vertices of Σ'_Q is not contained in P_I.

We now assume $w_P = w_Q$ and $w_Q \neq u$ and use w to denote w_Q. Let $a_Q^T x \leq \beta_Q$ be a face-defining inequality F_Q. Remember that $\nu(x^*,Q) = (u,-h,\nu(w,F_Q))$ and that $\nu(x^*,P) = (u,-h,\nu(w,F_P))$. The assertion then follows by induction if one can show that $F_P \subseteq F_Q$ holds.

Since $P_I \subseteq Q_I$ the inequality $a_Q^T x \leq \beta_Q$ is face-defining for P_I. The face F'_P of P_I, defined by this inequality contains w and thus F_P, by the minimality of F_P. The integer points in F'_P lie in P_I and hence in Q_I and they satisfy $a_Q^T x \leq \beta_Q$ with equality. Thus $F_P \subseteq F'_P \subseteq F_Q$. ∎

4.1 Details of Step 2 We now analyze the running time of the above procedure. We assume that P is represented by a system $Ax \leq b$, where $A \in \mathbb{Q}^{m\times n}$ and $b \in \mathbb{Q}^m$. Again, φ denotes the largest binary encoding length of a coefficient of A, b and x^*.

Step 1 of the above procedure takes an expected number of $O(m + \varphi \log m)$ 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 (For an alleged solution $(\bar{y}^T, \bar{z})$ compute $\max_i \bar{y}^T x_i$ and check the inequality. The d equalities can be checked easily). 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 [5] 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]. We look for a valid inequality that is tight at w with smallest possible dimension, i.e. the number of affinely independent points of P_I that are tight at w should be as small as possible. 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. A point in the relative interior of this set defines a face of minimal dimension that is 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.

4.2 The size of a basis 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. A smallest (number of constraints) subsystem $A'x \leq b'$ of $Ax \leq b$ with $\Sigma(x^*, Ax \leq b) = \Sigma(x^*, A'x \leq b')$ is called a *basis* of $Ax \leq b$. The goal of this section is to show, that the number of constraints of a basis of $Ax \leq b$ is bounded by a constant. The following theorem is due to Scarf [10], see also [11, p. 234].

THEOREM 4.1. *Let $Ax \leq b$ be a system of inequalities in d variables, and let $c \in \mathbb{R}^d$. If $\max\{c^T x \mid Ax \leq b,\ x \in \mathbb{Z}^d\}$ is finite, then there exists a subset $A'x \leq b'$ of $Ax \leq b$ with at most $2^d - 1$ inequalities, such that the* *following equality holds*

$$\max\{c^T x \mid Ax \le b,\, x \in \mathbb{Z}^d\}$$
$$= \max\{c^T x \mid A'x \le b',\, x \in \mathbb{Z}^d\}$$

From this one can immediately infer the next statement, which is useful in our setting.

COROLLARY 4.1. *Let $P = \{x \in \mathbb{R}^d \mid Ax \le b\}$ be a rational polyhedron and let $F_P \subseteq P_I$ be a face of P_I. Then there exists a subsystem $A'x \le b'$ of $Ax \le b$, which consists of at most $2\,d\,(2^d - 1)$ constraints, defining a polyhedron $Q = \{x \in \mathbb{R}^d \mid A'x \le b'\}$, such that there exists a face F_Q of Q_I with $F_P \subseteq F_Q$ and* $\dim(F_P) = \dim(F_Q)$. *Furthermore, an inequality $a^T x \le \beta$ is a face-defining inequality of F_P if and only if $a^T x \le \beta$ is a face-defining inequality of F_Q.*

Proof. The polyhedron P_I can be described as

$$P_I = \{x \in \mathbb{R}^d \mid A^{=}x = b^{=}\} \cap \{x \in \mathbb{R}^d \mid A^{+}x \le b^{+}\},$$

such that the following conditions hold, see [11, p. 103]: The matrix $A^{=}$ has full row-rank and d_1 rows, where $d - d_1$ is the dimension of P_I. Furthermore each constraint of $A^{+}x \le b^{+}$ is irredundant and each row of A^{+} is orthogonal to the rows of $A^{=}$.

A face F_P of P_I of dimension $k \le d - d_1$ is determined by $d - d_1 - k$ linearly independent constraints $\widetilde{A}x \le \widetilde{b}$ from the set $A^{+}x \le b^{+}$ which are satisfied by F_P with equality. It follows from Theorem 2 that there exists a subset $A'x \le b'$ of $Ax \le b$ with at most $2\,d\,(2^d - 1)$ constraints, defining a polyhedron Q, such that $A^{=}x = b^{=}$ and $\widetilde{A}x \le \widetilde{b}$ are valid for Q_I, where $Q = \{x \in \mathbb{R}^d \mid A'x \le b'\}$. The face F_Q of Q_I which results from setting the constraints in $\widetilde{A}x \le \widetilde{b}$ to equality contains F_P and has dimension k.

Furthermore, an inequality $a^T x \le \beta$ is a face-defining inequality of F_P or of F_Q if and only if $a = \mu\, A^{=} + \lambda\, \widetilde{A}$ and $\beta = \mu\, b + \lambda\, \widetilde{b}$, where $\mu \in \mathbb{R}^{d_1}$, $\lambda \in \mathbb{R}^{d - d_1 - k}$ and λ is strictly positive. ∎

This enables us to estimate the size of a basis of $Ax \le b$.

LEMMA 4.2. *Let $P = \{x \in \mathbb{R}^d \mid Ax \le b\}$ be a polytope in fixed dimension d with nonempty integer hull. Suppose that $\Sigma(x^*, Ax \le b)$ has k vertices. Then there exists a subset $A'x \le b'$ of the constraints $Ax \le b$, with at most $(2k-1)\,2\,d\,(2^d - 1)$ constraints, such that the simplices $\Sigma(x^*, Ax \le b)$ and $\Sigma(x^*, A'x \le b')$ are equal.*

Proof. We use induction on the number k. It follows from Corollary 1 that $2\,d\,(2^d - 1)$ constraints suffice to fix the lexicographically minimal vertex u of P_I and to detect the case, where $w = u$. Thus if $k = 1$ the assertion holds.

Suppose now that $k \ge 2$. Again, by Corollary 1, there exists a subset $A'x \le b'$ defining a polyhedron Q, such that the ray $\overrightarrow{u, x^*}$ leaves Q_I also in the point w and such that $a^T x \le \beta$ defines the minimal face of P_I containing w if and only if $a^T x \le \beta$ defines the minimal face of Q_I containing w.

This means that there exists a subset of $4\,d\,(2^d - 1)$ constraints with respect to which the first two steps of the procedure **ContainingSimplex** yields the same result. The assertion then follows by induction. ∎

4.3 The LP-type problem We now model the search for a constant size subset $A'x \le b'$ of $Ax \le b$, such that $\Sigma(x^*, Ax \le b) = \Sigma(x^*, A'x \le b')$ 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 $(\mathcal{W}, \le)$ 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. For a subset $F \subseteq H$, we denote by $P(F)$ the polytope which is defined by F and the implicit upper and lower bounds. We then define $\omega(F) \le \omega(G)$ if $P(F) \preceq_{x^*} P(H)$. The following axioms are satisfied.

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

This axiom is immediate by Lemma 1, since $P(G) \subseteq P(F)$.

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 $\Sigma(x^*, P(F))$ and $\Sigma(x^*, P(G))$ coincide. If $\omega(G) < \omega(G \cup \{h\})$, then h cuts off a vertex of this simplex (Lemma 1). 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 2, 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 $\omega(B) \ne \omega(B \cup \{h\})$. A basis computation for a set of constraints G is the task of computing a basis of G.

Let us 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 $\Sigma(x^*, P(B))$. 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 $\Sigma(x^*, P(F))$ for each $F \subseteq G$ and choose the smallest set F with $\Sigma(x^*, P(F)) = \Sigma(x^*, P(G))$. This can be done by calling our procedure **ContainingSimplex(G, x^*)** a constant number of times. These calls cost $\mathrm{O}(\varphi^2)$.

This shows that one can compute $\Sigma(x^*, Ax \leq b)$ and a basis $A'x \leq b'$ of $Ax \leq b$ in expected time $O(m + \varphi^2 \log m)$. Recall that $x^* \in P_I$ if and only if $x^* \in \Sigma(x^*, Ax \leq b)$ if and only if x^* is in the integer hull of $A'x \leq b'$. If x^* is not in $\Sigma(x^*, Ax \leq b)$ we can compute a separating hyperplane of x^* from the integer hull of $A'x \leq b'$ in $O(\varphi^2)$ steps, using the equivalence of separation and optimization.

All-together we have shown that the point containment problem in the integer hull of a polyhedron can be solved in an expected number of $O(m + \varphi^2 \log m)$ operations. This concludes the proof of Theorem 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.