MATHEMATICS OF OPERATIONS RESEARCH
Vol 8, No. 4, November 1983
Printed in U.S.A.

INTEGER PROGRAMMING WITH A FIXED NUMBER OF VARIABLES*

H. W. LENSTRA, JR.

Universiteit van Amsterdam

It is shown that the integer linear programming problem with a fixed number of variables is polynomially solvable. The proof depends on methods from geometry of numbers.

The *integer linear programming problem* is formulated as follows. Let n and m be positive integers, A an $m \times n$-matrix with integral coefficients, and $b \in \mathbb{Z}^m$. The question is to decide whether there exists a vector $x \in \mathbb{Z}^n$ satisfying the system of m inequalities $Ax \leqslant b$. No algorithm for the solution of this problem is known which has a running time that is bounded by a polynomial function of the *length* of the data. This length may, for our purposes, be defined to be $n \cdot m \cdot \log(a + 2)$, where a denotes the maximum of the absolute values of the coefficients of A and b. Indeed, no such *polynomial algorithm* is likely to exist, since the problem in question is *NP-complete* [3], [12].

In this paper we consider the integer linear programming problem with a fixed value of n. In the case $n = 1$ it is trivial to design a polynomial algorithm for the solution of the problem. For $n = 2$, Hirschberg and Wong [5] and Kannan [6] have given polynomial algorithms in special cases. A complete treatment of the case $n = 2$ was given by Scarf [10]. It was conjectured [5], [10] that for any fixed value of n there exists a polynomial algorithm for the solution of the integer linear programming problem. In the present paper we prove this conjecture by exhibiting such an algorithm. The degree of the polynomial by which the running time of our algorithm can be bounded is an exponential function of n.

Our algorithm is described in §1. Using tools from geometry of numbers [1] we show that the problem can be transformed into an equivalent one having the following additional property: either the existence of a vector $x \in \mathbb{Z}^n$ satisfying $Ax \leqslant b$ is obvious; or it is known that the last coordinate of any such x belongs to an interval whose length is bounded by a constant only depending on n. In the latter case, the problem is reduced to a bounded number of lower dimensional problems.

If in the original problem each coordinate of x is required to be in $\{0, 1\}$, no transformation of the problem is needed to achieve the condition just stated. This suggests that in this case our algorithm is equivalent to complete enumeration. We remark that the $\{0, 1\}$ linear programming problem is *NP*-complete.

In the general case we need two auxiliary algorithms for the construction of the required transformation. The first of these, which "remodels" the convex set $\{x \in \mathbb{R}^n : Ax \leqslant b\}$, is given in §2. L. Lovász observed that my original algorithm for this could be made polynomial even for varying n, by employing the polynomial solvability of the linear programming problem [8], [4]. I am indebted to Lovász for permission to describe the improved algorithm in §2.

*Received November 13, 1981; revised July 2, 1982.
AMS 1980 subject classification. Primary: 68C25; Secondary: 90C10.
OR/MS Index 1978 subject classification. Primary: 625 Programming/integer/algorithms.
Key words. Integer programming, polynomial algorithm, geometry of numbers.

0364-765X/83/0804/0538$01.25

The second auxiliary algorithm is a reduction process for n-dimensional lattices. Such an algorithm, also due to Lovász, appeared in [9, §1], and a brief sketch is given in §3 of the present paper. This algorithm is polynomial even for varying n. It supersedes the much inferior algorithm that was described in an earlier version of this paper.

In §4 we prove, following a suggestion of P. van Emde Boas, that the integer linear programming problem with a fixed value of m is also polynomially solvable. This is an immediate consequence of our main result.

§5 is devoted to the *mixed integer linear programming problem*. Combining our methods with Khachiyan's results [8], [4] we show that this problem is polynomially solvable for any fixed value of the number of integer variables. This generalizes both our main result and Khachiyan's theorem.

The algorithms presented in this paper were designed for theoretical purposes only, and there are several modifications that might improve their practical performance. It is to be expected that the practical value of our algorithms is restricted to small values of n.

It is a pleasure to acknowledge my indebtedness to P. van Emde Boas, not only for permission to include §4, but also for suggesting the problem solved in this paper and for several inspiring and stimulating discussions.

1. Description of the algorithm. Let K denote the closed convex set

$$K = \{x \in \mathbb{R}^n : Ax \leqslant b\}$$

The question to be decided is whether $K \cap \mathbb{Z}^n = \emptyset$. In the description of the algorithm that follows, we make the following two simplifying assumptions about K:

(1) K is *bounded*;

(2) K has *positive volume*.

The first assumption is justified by the following result, which is obtained by combining a theorem of Von zur Gathen and Sieveking [12] with Hadamard's determinant inequality (cf. (6) below): the set $K \cap \mathbb{Z}^n$ is nonempty if and only if $K \cap \mathbb{Z}^n$ contains a vector whose coefficients are bounded by $(n+1)n^{n/2}a^n$ in absolute value, where a is as in the introduction. Adding these inequalities to the system makes K bounded.

For the justification of condition (2) we refer to §2. Under the assumptions (1) and (2), §2 describes how to construct a nonsingular endomorphism τ of the vector space $\mathbb{R}^n$, such that τK has a "spherical" appearance. More precisely, let $|\ |$ denote the Euclidean length in $\mathbb{R}^n$, and put

$$B(p,z) = \{x \in \mathbb{R}^n : |x-p| \leqslant z\} \qquad \text{for} \quad p \in \mathbb{R}^n, \quad z \in \mathbb{R}_{>0},$$

the closed ball with center p and radius z. With this notation, the τ constructed will satisfy

$$B(p,r) \subset \tau K \subset B(p,R) \tag{3}$$

for some $p \in \tau K$, with r and R satisfying

$$\frac{R}{r} \leqslant c_1, \tag{4}$$

where c_1 is a constant only depending on n.

Let such a τ be fixed, and put $L = \tau\mathbb{Z}^n$. This is a *lattice* in $\mathbb{R}^n$, i.e., there exists a basis $b_1, b_2, \ldots, b_n$ of $\mathbb{R}^n$ such that

$$L = \sum_{i=1}^{n} \mathbb{Z}b_i = \left\{ \sum_{i=1}^{n} m_i b_i : m_i \in \mathbb{Z}\ (1 \leqslant i \leqslant n) \right\} \tag{5}$$

We can take, for example, $b_i = \tau(e_i)$, with e_i denoting the ith standard basis vector of $\mathbb{R}^n$. We call $b_1, b_2, \ldots, b_n$ a *basis* for L if (5) holds. If $b'_1, b'_2, \ldots, b'_n$ is another basis for L, then $b'_i = \sum_{j=1}^{n} m_{ij} b_j$ for some $n \times n$-matrix $M = (m_{ij})_{1 \leqslant i,j \leqslant n}$ with integral coefficients and $\det(M) = \pm 1$. It follows that the positive real number $|\det(b_1, b_2, \ldots, b_n)|$ (the b_i being written as column vectors) only depends on L, and not on the choice of the basis; it is called the *determinant* of L, notation: $d(L)$. We can interpret $d(L)$ as the volume of the parallelepiped $\sum_{i=1}^{n} [0,1) \cdot b_i$, where $[0,1) = \{z \in \mathbb{R} : 0 \leqslant z < 1\}$. This interpretation leads to the *inequality of Hadamard*

$$d(L) \leqslant \prod_{i=1}^{n} |b_i|. \tag{6}$$

The equality sign holds if and only if the basis $b_1, b_2, \ldots, b_n$ is orthogonal. It is a classical theorem that L has a basis $b_1, b_2, \ldots, b_n$ that is nearly orthogonal in the sense that the following inequality holds:

$$\prod_{i=1}^{n} |b_i| \leqslant c_2 \cdot d(L) \tag{7}$$

where c_2 is a constant only depending on n, cf. [1, Chapter VIII], [11]. In §3 we shall indicate a *reduction process*, i.e., an algorithm that changes a given basis for L into one satisfying (7).

LEMMA. *Let $b_1, b_2, \ldots, b_n$ be any basis for L. Then*

$$\forall x \in \mathbb{R}^n : \exists y \in L : |x - y|^2 \leqslant \tfrac{1}{4}(|b_1|^2 + \cdots + |b_n|^2). \tag{8}$$

PROOF. We use induction on n, the case $n = 1$ (or $n = 0$) being obvious. Let $L' = \sum_{i=1}^{n-1} \mathbb{Z} b_i$, this is a lattice in the $(n-1)$-dimensional hyperplane $H = \sum_{i=1}^{n-1} \mathbb{R} b_i$. Denote by h the distance of b_n to H. Clearly we have

$$h \leqslant |b_n|. \tag{9}$$

Now to prove (8), let $x \in \mathbb{R}^n$. We can find $m \in \mathbb{Z}$ such that the distance of $x - mb_n$ to H is $\leqslant \frac{1}{2}h$. Write $x - mb_n = x_1 + x_2$, with $x_1 \in H$ and x_2 perpendicular to H. Then $|x_2| \leqslant \frac{1}{2}h \leqslant \frac{1}{2}|b_n|$. By the induction hypothesis there exists $y_1 \in L'$ such that $|x_1 - y_1|^2 \leqslant \frac{1}{4}(|b_1|^2 + \cdots + |b_{n-1}|^2)$. Since x_2 is orthogonal to y_1 the element $y = y_1 + mb_n$ of L now satisfies $|x - y|^2 = |x_1 - y_1|^2 + |x_2|^2 \leqslant \frac{1}{4}(|b_1|^2 + \cdots + |b_{n-1}|^2 + |b_n|^2)$. This proves the lemma.

Notice that the proof gives an effective construction of the element $y \in L$ that is asserted to exist.

If we number the b_i such that $|b_n| = \max\{|b_i| : 1 \leqslant i \leqslant n\}$, then (8) implies

$$\forall x \in \mathbb{R}^n : \exists y \in L : |x - y| \leqslant \tfrac{1}{2}\sqrt{n}\,|b_n|. \tag{10}$$

Now assume that $b_1, b_2, \ldots, b_n$ is a *reduced* basis for L in the sense that (7) holds, and let L' and h have the same meaning as in the proof of the lemma. It is easily seen that

$$d(L) = h \cdot d(L'). \tag{11}$$

From (7), (11) and (6), applied to L', we get

$$\prod_{i=1}^{n} |b_i| \leqslant c_2 \cdot d(L) = c_2 \cdot h \cdot d(L') \leqslant c_2 \cdot h \cdot \prod_{i=1}^{n-1} |b_i|$$

and therefore, with (9):

$$c_2^{-1}\ |b_n| \leqslant h \leqslant |b_n|. \tag{12}$$

After these preparations we describe the procedure by which we decide whether $K \cap \mathbb{Z}^n = \emptyset$ or, equivalently, $\tau K \cap L = \emptyset$. We assume that $b_1, b_2, \ldots, b_n$ is a basis for L for which (7) holds, numbered such that $|b_n| = \max\{|b_i| : 1 \leqslant i \leqslant n\}$.

Applying (10) with $x = p$ we find a vector $y \in L$ with $|p - y| \leqslant \frac{1}{2}\sqrt{n}\,|b_n|$. If $y \in \tau K$ then $\tau K \cap L \neq \emptyset$, and we are done. Suppose therefore that $y \notin \tau K$. Then $y \notin B(p,r)$, by (3), so $|p - y| > r$, and this implies that $r < \frac{1}{2}\sqrt{n}\,|b_n|$. Let now H, L', h have the same meaning as in the proof of the lemma. We have

$$L = L' + \mathbb{Z}b_n \subset H + \mathbb{Z}b_n = \bigcup_{k \in \mathbb{Z}} (H + kb_n).$$

Hence L is contained in the union of countably many parallel hyperplanes, which have successive distances h from each other. We are only interested in those hyperplanes that have a nonempty intersection with τK, these have, by (3), also a nonempty intersection with $B(p,R)$. Suppose that precisely t of the hyperplanes $H + kb_n$ intersect $B(p,R)$. Then we have clearly $t - 1 \leqslant 2R/h$. By (4) and (12) we have

$$2R \leqslant 2rc_1 < c_1\sqrt{n}\,|b_n|, \qquad h \geqslant c_2^{-1}|b_n|$$

so $t - 1 < c_1c_2\sqrt{n}$. Hence the number of values for k that have to be considered is bounded by a constant only depending on n. Which values of k need be considered can easily be deduced from a representation of p as a linear combination of $b_1, b_2, \ldots, b_n$.

If we fix the value of k then we restrict attention to those $x = \sum_{i=1}^{n} y_i b_i$ for which $y_n = k$; and this leads to an integer programming problem with $n - 1$ variables $y_1, y_2, \ldots, y_{n-1}$. It is straightforward to show that the length of the data of this new problem is bounded by a polynomial function of the length of the original data, if the directions of §2 have been followed for the construction of τ.

Each of the lower dimensional problems is treated recursively. The case of dimension $n = 1$ (or even $n = 0$) may serve as a basis for the recursion. This finishes our description of the algorithm.

We observe that in the case that $K \cap \mathbb{Z}^n$ is nonempty, our algorithm actually produces an element $x \in K \cap \mathbb{Z}^n$.

2. The convex set K. Let $K = \{x \in \mathbb{R}^n : Ax \leqslant b\}$, and assume that K is bounded. In this section we describe an algorithm that can be used to verify that K satisfies condition (2) of §1; to reduce the number of variables if that condition is found not to be satisfied; and to find the map τ used in §1. The algorithm is better than what is strictly needed in §1, in the sense that it is polynomial even for varying n. I am indebted to L. Lovász for pointing out to me how this can be achieved.

In the first stage of the algorithm one attempts to construct vertices $v_0, v_1, \ldots, v_n$ of K whose convex hull is an n-simplex of *positive* volume. By maximizing an arbitrary linear function on K, employing Khachiyan's algorithm [8], [4], one finds a vertex v_0 of K, unless K is empty. Suppose, inductively, that vertices $v_0, v_1, \ldots, v_d$ of K have been found for which $v_1 - v_0, \ldots, v_d - v_0$ are linearly independent, with $d < n$. Then we can construct $n - d$ linearly independent linear functions $f_1, \ldots, f_{n-d}$ on $\mathbb{R}^n$ such that the d-dimensional subspace $V = \sum_{j=1}^{d} \mathbb{R}(v_j - v_0)$ is given by

$$V = \{x \in \mathbb{R}^n : f_1(x) = \cdots = f_{n-d}(x) = 0\}.$$

Again employing Khachiyan's algorithm, we maximize each of the linear functions $f_1, -f_1, f_2, -f_2, \ldots, f_{n-d}, -f_{n-d}$ on K, until a vertex v_{d+1} of K is found for which $f_j(v_{d+1}) \neq f_j(v_0)$ for some $j \in \{1, 2, \ldots, n-d\}$. If this occurs, then $v_1 - v_0, \ldots, v_d - v_0, v_{d+1} - v_0$ are linearly independent, and the inductive step of the construction is completed. If, on the other hand, no such v_{d+1} is found after each of the $2(n-d)$ functions $f_1, -f_1, \ldots, f_{n-d}, -f_{n-d}$ has been maximized, then we must have $f_j(x) = f_j(v_0)$ for all $x \in K$ and all $j = 1, 2, \ldots, n-d$, and therefore $K \subset v_0 + V$. In this case we reduce the problem to an integer programming problem with only d variables, as follows.

Choose, for $j = 1, 2, \ldots, d$, a nonzero scalar multiple w_j of $v_j - v_0$ such that $w_j \in \mathbb{Z}^n$, and denote by W the $(n \times d)$-matrix whose columns are the w_j. Notice that W has rank d. Employing the *Hermite normal form* algorithm of Kannan and Bachem [7] we can find, in polynomial time, an integral $n \times n$-matrix U with $\det(U) = \pm 1$ such that

$$UW = (k_{ij})_{1 \leqslant i \leqslant n,\, 1 \leqslant j \leqslant d}$$

with

$$\begin{cases} k_{ij} = 0 & \text{if} \quad i > j, \\ k_{ii} \neq 0 & \text{for} \quad 1 \leqslant i \leqslant d. \end{cases} \tag{13}$$

Denote by $u_1, u_2, \ldots, u_n$ the columns of the integral matrix U^{-1}. These form a basis of $\mathbb{R}^n$, and also of the lattice $\mathbb{Z}^n$: $\mathbb{Z}^n = \sum_{j=1}^n \mathbb{Z} u_j$. The subspace V of $\mathbb{R}^n$ is generated by the columns of $W = U^{-1} \cdot (k_{ij})$, so (13) implies that

$$V = \sum_{j=1}^{d} \mathbb{R} u_j . \tag{14}$$

Define $r_1, r_2, \ldots, r_n \in \mathbb{R}$ by $v_0 = \sum_{j=1}^n r_j u_j$; so $(r_j)_{j=1}^n = U v_0$.

Now suppose that $x \in K \cap \mathbb{Z}^n$. Then $x = \sum_{j=1}^n y_j u_j$ with $y_j \in \mathbb{Z}$, and $x \in K$ implies that $x - v_0 \in V$. By (14) this means that $y_j = r_j$ for $d < j \leqslant n$. So if at least one of $r_{d+1}, \ldots, r_n$ is not an integer, then $K \cap \mathbb{Z}^n = \emptyset$. Suppose, therefore, that $r_{d+1}, \ldots, r_n$ are all integral. Substituting $x = \sum_{j=1}^d y_j u_j + \sum_{j=d+1}^n r_j u_j$ in our original system $Ax \leqslant b$ we then see that the problem is equivalent to an integer programming problem with d variables $y_1, y_2, \ldots, y_d$, as required. The vertices $v_0, v_1, \ldots, v_d$ of K give rise to $d+1$ vertices $v'_0, v'_1, \ldots, v'_d$ of the convex set in $\mathbb{R}^d$ belonging to the new problem, and $v'_0, v'_1, \ldots, v'_d$ span a d-dimensional simplex of positive volume. This means that for the new, d-dimensional problem the first stage of the algorithm that we are describing can be bypassed.

To conclude the first stage of the algorithm, we may now suppose that for each $d = 0, 1, \ldots, n-1$ the construction of v_{d+1} is successful. Then after n steps we have $n+1$ vertices $v_0, v_1, \ldots, v_n$ of K for which $v_1 - v_0, \ldots, v_n - v_0$ are linearly independent. The n-simplex spanned by $v_0, v_1, \ldots, v_n$ is contained in K, and its volume equals $|\det M|/n!$ where M is the matrix with column vectors $v_1 - v_0, \ldots, v_n - v_0$. This is positive, so condition (2) of §1 is satisfied.

In the second stage of the algorithm we construct the coordinate transformation τ needed in §1. To this end we first try to find a simplex of "large" volume in K. This is done by an iterative application of the following procedure, starting from the simplex spanned by $v_0, v_1, \ldots, v_n$. The volume of that simplex is denoted by $\mathrm{vol}(v_0, v_1, \ldots, v_n)$.

Construct $n+1$ linear functions $g_0, g_1, \ldots, g_n : \mathbb{R}^n \to \mathbb{R}$ such that

$$\begin{aligned} &g_i \text{ is constant on } \{v_j : 0 \leqslant j \leqslant n, j \neq i\}, \\ &g_i(v_i) \neq g_i(v_j) \qquad \text{for} \quad 0 \leqslant j \leqslant n, j \neq i, \end{aligned} \tag{15}$$

for $i = 0, 1, \ldots, n$. Maximizing the functions $g_0, -g_0, g_1, -g_1, \ldots, g_n, -g_n$ on K by Khachiyan's algorithm we can decide whether there exist $i \in \{0, 1, \ldots, n\}$ and a vertex x of K such that

$$|g_i(x - v_j)| > \tfrac{3}{2}|g_i(v_i - v_j)|$$

for $j \neq i$ (the choice of j is immaterial, by (15)).

Suppose that such a pair i, x is found. Then we replace v_i by x. This replacement enlarges $\mathrm{vol}(v_0, v_1, \ldots, v_n)$ by a factor $|g_i(x - v_j)|/|g_i(v_i - v_j)|$ (for $j \neq i$), which is more than 3/2. We now return to the beginning of the procedure ("Construct $n + 1$ linear functions . . . ").

In every iteration step $\mathrm{vol}(v_0, v_1, \ldots, v_n)$ increases by a factor $> 3/2$. On the other hand, this volume is bounded by the volume of K. Hence after a polynomially bounded number of iterations we reach a situation in which the above procedure discovers that

$$|g_i(x - v_j)| \leqslant \tfrac{3}{2}|g_i(v_i - v_j)| \tag{16}$$

for all $x \in K$ and all $i, j \in \{0, 1, \ldots, n\}$ with $i \neq j$. In that case we let τ be a nonsingular endomorphism of $\mathbb{R}^n$ with the property that $\tau(v_0), \tau(v_1), \ldots, \tau(v_n)$ span a *regular* n-simplex. With $p = (n+1)^{-1}\sum_{j=0}^{n}\tau(v_j)$ we now claim that $B(p, r) \subset \tau K \subset B(p, R)$ for certain positive real numbers r, R satisfying $R/r \leqslant 2n^{3/2}$, i.e., that conditions (3) and (4) of §1 are satisfied, with $c_1 = 2n^{3/2}$. This finishes the description of our algorithm.

To prove our claim, we write $z_j = \tau(v_j)$, for $0 \leqslant j \leqslant n$; we write S for the regular n-simplex spanned by $z_0, z_1, \ldots, z_n$, and we define, for $c \geqslant 1$:

$$T_c = \{x \in \mathbb{R}^n : \mathrm{vol}(z_0, \ldots, z_{i-1}, x, z_{i+1}, \ldots, z_n) \leqslant c \cdot \mathrm{vol}(z_0, \ldots, z_n) \text{ for all } i \in \{0, 1, \ldots, n\}\}.$$

Condition (16) (for all $x \in K$ and all $i \neq j$) means precisely that $\tau K \subset T_{3/2}$. Further, it is clear that $S \subset \tau K$. Our claim now follows from the following lemma.

LEMMA. *Let $c \geqslant 1$. With the above notation we have $B(p, r) \subset S \subset T_c \subset B(p, R)$ for two positive real numbers r, R satisfying*

$$\left(\frac{R}{r}\right)^2 = \begin{cases} c^2n^3 + (c^2+1)n^2 & \text{if } n \text{ is even,} \\ c^2n^3 + (2c^2 - 2c + 1)n^2 + (c^2 - 2c)n & \text{if } n \text{ is odd.} \end{cases}$$

PROOF. Using a similarity transformation we can identify $\mathbb{R}^n$ with the hyperplane $\{(r_j)_{j=0}^n \in \mathbb{R}^{n+1} : \sum_{j=0}^n r_j = 1\}$ in $\mathbb{R}^{n+1}$ such that $z_0, z_1, \ldots, z_n$ is the standard basis of $\mathbb{R}^{n+1}$. Then we have

$$p = \frac{1}{n+1}\sum_{j=0}^{n} z_j = \left(\frac{1}{n+1}, \frac{1}{n+1}, \ldots, \frac{1}{n+1}\right),$$

and

$$T_c = \left\{(r_j)_{j=0}^n \in \mathbb{R}^{n+1} : |r_j| \leqslant c \text{ for } 0 \leqslant j \leqslant n, \text{ and } \sum_{j=0}^{n} r_j = 1\right\}.$$

By a straightforward analysis one proves that T_c is the convex hull of the set of points obtained by permuting the coordinates of the point

$$z_0 - c\sum_{j=1}^{m} z_j + c\sum_{j=m+1}^{n} z_j \qquad \text{if} \quad n = 2m,$$

$$(1-c)z_0 - c\sum_{j=1}^{m} z_j + c\sum_{j=m+1}^{n} z_j \qquad \text{if} \quad n = 2m+1.$$

It follows that $T_c \subset B(p,R)$, where R is the distance of p to the above point:

$$R^2 = \begin{cases} nc^2 + \dfrac{n}{n+1} & \text{if } n \text{ is even,} \\ (n+1)c^2 - 2c + \dfrac{n}{n+1} & \text{if } n \text{ is odd.} \end{cases}$$

Further, $B(p,r) \subset S$, where r is the distance of p to $(0, 1/n, 1/n, \ldots, 1/n)$:

$$r^2 = \frac{1}{n(n+1)}.$$

This proves the lemma.

REMARKS. (a) To the construction of τ in the above algorithm one might raise the objection that τ need not be given by a matrix with *rational* coefficients. Indeed, for $n = 2, 4, 5, 6, 10, \ldots$ there exists no regular n-simplex all of whose vertices have rational coordinates. This objection can be answered in several ways. One might replace the regular simplex by a rational approximation of it, or indeed by any fixed n-simplex with rational vertices and positive volume, at the cost of getting a larger value for c_1. Alternatively, one might embed $\mathbb{R}^n$ in $\mathbb{R}^{n+1}$, as was done in the proof of the lemma. Finally, it can be argued that it is not necessary that the matrix M_τ defining τ be rational, but only the symmetric matrix $M_\tau{}^{\top}M_\tau$ defining the quadratic form $(\tau x, \tau x)$; and this can easily be achieved in the above construction of τ.

(b) The proof that the algorithm described in this section is polynomial, even for varying n, is entirely straightforward. We indicate the main points. The construction of $f_1, \ldots, f_{n-d}$ in the first stage, and of $g_0, g_1, \ldots, g_n$ in the second stage, can be done by Gaussian elimination, which is well known to be a polynomial algorithm, cf. [2, §7]. It follows that Khachiyan's algorithm is only applied to problems whose lengths are bounded by a polynomial function of the length of the original data. The same applies to the d-dimensional integer programming problem constructed in the first stage. Further details are left to the reader.

(c) We discuss to which extent the value $2n^{3/2}$ for c_1 in (4) is best possible. Replacing the coefficient 3/2 in (16) by other constants $c > 1$ we find, using the lemma, that for any fixed $\epsilon > 0$ we can take

$$c_1 = \begin{cases} \left((1+\epsilon)(n^3 + 2n^2)\right)^{1/2} & \text{if } n \text{ is even,} \\ \left((1+\epsilon)(n^3 + n^2 - n)\right)^{1/2} & \text{if } n \text{ is odd.} \end{cases}$$

If one is satisfied with an algorithm that is only polynomial for fixed n one can also take $\epsilon = 0$ in this formula. To achieve this, one uses a list of all vertices of K to find the simplex of maximal volume inside K, and transforms this simplex into a regular one. The following result shows that there is still room for improvement: if $K \subset \mathbb{R}^n$ is any closed convex set satisfying (1) and (2) then there exists a nonsingular endomorphism τ of $\mathbb{R}^n$ such that (3) and (4) hold with $c_1 = n$. To prove this, one chooses an *ellipsoid* E inside K with maximal volume, and one chooses τ such that τE is a sphere. The case that K is a simplex shows that the value $c_1 = n$ is best possible. For fixed n and $\epsilon > 0$ there is a polynomial algorithm that achieves $c_1 = (1+\epsilon)n$. I do not know how well the best possible value $c_1 = n$ can be approximated by an algorithm that is polynomial for varying n.

(d) The algorithm described in this section applies equally well to any class $\mathcal{K}$ of compact convex bodies in $\mathbb{R}^n$ for which there exists a polynomial algorithm that maximizes linear functions on members K of $\mathcal{K}$. This remark will play an important role in §5. In particular, we can take for $\mathcal{K}$ a "solvable" class of convex bodies, in the terminology of [4, §§1 and 3]. The same remark can be made for the algorithm presented in §1.

3. The reduction process. Let n be a positive integer, and let $b_1, b_2, \ldots, b_n \in \mathbb{R}^n$ be n linearly independent vectors. Put $L = \sum_{i=1}^n \mathbb{Z} b_i$; this is a lattice in $\mathbb{R}^n$. In this section we indicate an algorithm that transforms the basis $b_1, b_2, \ldots, b_n$ for L into one satisfying (7) with $c_2 = 2^{n(n-1)/4}$ The algorithm is taken from [9, §1], to which we refer for a more detailed description.

We recall the Gram-Schmidt orthogonalization process. The vectors b_i^* $(1 \leqslant i \leqslant n)$ and the real numbers μ_{ij} $(1 \leqslant j < i \leqslant n)$ are inductively defined by

$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^*, \qquad \mu_{ij} = (b_i, b_j^*)/(b_j^*, b_j^*),$$

where (,) denotes the ordinary inner product on $\mathbb{R}^n$. Notice that b_i^* is the projection of b_i on the orthogonal complement of $\sum_{j=1}^{i-1} \mathbb{R} b_j$, and that $\sum_{j=1}^{i-1} \mathbb{R} b_j = \sum_{j=1}^{i-1} \mathbb{R} b_j^*$, for $1 \leqslant i \leqslant n$. It follows that $b_1^*, b_2^*, \ldots, b_n^*$ is an orthogonal basis of $\mathbb{R}^n$. The following result is taken from [9].

PROPOSITION. *Suppose that*

$$|\mu_{ij}| \leqslant \tfrac{1}{2} \tag{17}$$

for $1 \leqslant j < i \leqslant n$, *and*

$$|b_i^* + \mu_{i\,i-1} b_{i-1}^*|^2 \geqslant \tfrac{3}{4} |b_{i-1}^*|^2 \tag{18}$$

for $1 < i \leqslant n$. *Then*

$$\prod_{i=1}^{n} |b_i| \leqslant 2^{n(n-1)/4} d(L),$$

i.e., (7) *holds with* $c_2 = 2^{n(n-1)/4}$.

PROOF. See [9, Proposition 1.6].

To explain condition (18) we remark that the vectors $b_i^* + \mu_{i\,i-1} b_{i-1}^*$ and b_{i-1}^* are the projections of b_i and b_{i-1} on the orthogonal complement of $\sum_{j=1}^{i-2} \mathbb{R} b_j$. Hence if (18) does not hold for some i, then it does hold for the basis obtained from $b_1, b_2, \ldots, b_n$ by interchanging b_{i-1} and b_i.

To change a given basis $b_1, b_2, \ldots, b_n$ for L into one satisfying (7) we may now iteratively apply the following transformations.

First transformation: select i, $1 < i \leqslant n$, such that (18) does not hold, and interchange b_{i-1} and b_i;

Second transformation: select i, j, $1 \leqslant j < i \leqslant n$, such that (17) does not hold, and replace b_i by $b_i - r b_j$, where r is the integer nearest to μ_{ij}.

It can be shown that, independently of the order in which these transformations are applied and independently of the choices of i, and of i and j, that are made, this leads after a finite number of steps to a basis $b_1, b_2, \ldots, b_n$ satisfying (17) and (18). Then (7) is satisfied as well, by the proposition. This finishes our sketch of the algorithm.

A particularly efficient strategy for choosing which transformation to apply, and for which i, or i and j, is described in [9, (1.15)]. If we assume the b_i to have *integer* coordinates then the resulting algorithm is polynomial, even for varying n, by [9, Proposition 1.26]. It follows that the same result is true if we allow the coordinates of the b_i to be *rational*.

REMARKS. (a) The algorithm sketched above can be used to find the shortest nonzero vector in L, in the following way. Suppose that $b_1, b_2, \ldots, b_n$ is a basis for L satisfying (7), and let $x \in L$. Then we can write $x = \sum_{i=1}^{n} m_i b_i$ with $m_i \in \mathbb{Z}$, and from Cramer's rule it is easy to derive that $|m_i| \leqslant c_2 \cdot |x|/|b_i|$, for $1 \leqslant i \leqslant n$. If x is the shortest nonzero vector in L then $|x| \leqslant |b_i|$ for all i, so $|m_i| \leqslant c_2$. So by searching the set $\{\sum_{i=1}^{n} m_i b_i : m_i \in \mathbb{Z},\ |m_i| \leqslant c_2 \text{ for } 1 \leqslant i \leqslant n\}$ we can find the shortest nonzero vector in L in polynomial time, for fixed n. For variable n this problem is likely to be NP-hard.

(b) We discuss to which extent our value for c_2 is best possible. The above algorithm yields $c_2 = 2^{n(n-1)/4}$. We indicate an algorithm that leads to a much better value for c_2; but the algorithm is only polynomial for fixed n.

In (a) we showed how to find the shortest nonzero vector in L by a search procedure. By an analogous but somewhat more complicated search procedure we can determine the *successive minima* $|b_1'|, |b_2'|, \ldots, |b_n'|$ of L (see [1, Chapter VIII] for the definition). Here $b_1', b_2', \ldots, b_n' \in L$ are linearly independent, and by [1, Chapter VIII, Theorem I, p. 205 and Chapter IV, Theorem VII, p. 120] they satisfy

$$\prod_{i=1}^{n} |b_i'| \leqslant \gamma_n^{n/2} \cdot d(L)$$

where γ_n denotes Hermite's constant [1, §IX.7, p. 247], for which it is known that

$$\frac{1}{2\pi e} + o(1) \leqslant \gamma_n/n \leqslant \frac{1}{\pi e} + o(1) \qquad \text{for} \quad n \to \infty.$$

Using a slight improvement of [1, Chapter V, Lemma 8, p. 135] we can change $b_1', b_2', \ldots, b_n'$ into a *basis* $b_1'', b_2'', \ldots, b_n''$ for L satisfying

$$|b_i''| \leqslant \max\left\{1, \tfrac{1}{2}\sqrt{i}\right\} \cdot |b_i'| \qquad (1 \leqslant i \leqslant n)$$

so

$$\prod_{i=1}^{n} |b_i''| \leqslant 2^{-n+2} \cdot \left(\tfrac{2}{3} n!\right)^{1/2} \cdot \gamma_n^{n/2} \cdot d(L) \qquad (\text{for } n \geqslant 3).$$

We conclude that, for fixed n, the basis $b_1, b_2, \ldots, b_n$ produced by the algorithm indicated in this section can be used to find, in polynomial time, a new basis satisfying (7), but now with $c_2 = (c \cdot n)^n$. Here c denotes some absolute positive constant.

On the other hand, the definition of γ_n implies that there exists an n-dimensional lattice L such that $|x| \geqslant \gamma_n^{1/2} \cdot d(L)^{1/n}$ for all $x \in L$, $x \neq 0$, cf. [1, Chapter I, Lemma 4, p. 21]. Any basis $b_1, b_2, \ldots, b_n$ for such a lattice clearly satisfies

$$\prod_{i=1}^{n} |b_i| \geqslant \gamma_n^{n/2} \cdot d(L).$$

Therefore the best possible value for c_2 satisfies $c_2 > (c' \cdot n)^{n/2}$ for some absolute positive constant c'.

4. A fixed number of constraints. In this section we show that the integer linear programming problem with a fixed value of m is polynomially solvable. It was noted by P. van Emde Boas that this is an immediate consequence of our main result.

Let n, m, A, b be as in the introduction. We have to decide whether there exists $x \in \mathbb{Z}^n$ for which $Ax \leqslant b$. Applying the algorithms of Kannan and Bachem [7] we can find an $(n \times n)$-matrix U with integral coefficients and determinant ± 1 such that the matrix $AU = (a_{ij}')_{1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n}$ satisfies

$$a_{ij}' = 0 \qquad \text{for} \quad j > i. \tag{19}$$

Putting $y = U^{-1}x$ we see that the existence of $x \in \mathbb{Z}^n$ with $Ax \leqslant b$ is equivalent to the existence of $y \in \mathbb{Z}^n$ with $(AU)y \leqslant b$. If $n > m$, then the coordinates $y_{m+1}, \ldots, y_n$ of y do not occur in these inequalities, since (19) implies that $a'_{ij} = 0$ for $j > m$. We conclude that the original problem can be reduced to a problem with only $\min\{n, m\}$ variables. The latter problem is, for fixed m, polynomially solvable, by the main result of this paper.

5. Mixed integer linear programming. The *mixed integer linear programming problem* is formulated as follows. Let k and m be positive integers, and n an integer satisfying $0 \leqslant n \leqslant k$. Let further A be an $m \times k$-matrix with integral coefficients, and $b \in \mathbb{Z}^m$. The question is to decide whether there exists a vector $x = (x_1, x_2, \ldots, x_k)^\top$ with

$$x_i \in \mathbb{Z} \quad \text{for} \quad 1 \leqslant i \leqslant n,$$

$$x_i \in \mathbb{R} \quad \text{for} \quad n+1 \leqslant i \leqslant k$$

satisfying the system of m inequalities $Ax \leqslant b$.

In this section we indicate an algorithm for the solution of this problem that is polynomial for any fixed value of n, the number of integer variables. This generalizes both the result of §1 ($n = k$) and the result of Khachiyan [8], [4] ($n = 0$).

Let

$$K' = \{x \in \mathbb{R}^k : Ax \leqslant b\},$$

$$K = \{(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n : \text{there exist } x_{n+1}, \ldots, x_k \in \mathbb{R} \text{ such that } (x_1, x_2, \ldots, x_k) \in K'\}.$$

The question is whether $K \cap \mathbb{Z}^n = \emptyset$.

Making use of the arguments of Von zur Gathen and Sieveking [12] we may again assume that K', and hence K, is bounded. Next we apply the algorithm of §2 to the compact convex set $K \subset \mathbb{R}^n$. To see that this can be done it suffices to show that we can maximize linear functions on K, see §2, Remark (d). But maximizing linear functions on K is equivalent to maximizing, on K', linear functions that depend only on the first n coordinates $x_1, x_2, \ldots, x_n$; and this can be done by Khachiyan's algorithm.

The rest of the algorithm proceeds as before. At a certain point in the algorithm we have to decide whether a given vector $y \in \mathbb{R}^n$ belongs to τK. This can be done by solving a linear programming problem with $k - n$ variables. This finishes the description of the algorithm.

As in §4 it can be proved that the mixed integer linear programming problem is also polynomially solvable if the number of inequalities that involve one or more integer variables is fixed; or, more generally, if the rank of the matrix formed by the first n columns of A is bounded.

References

[1] Cassels, J. W. S. (1959). *An Introduction to the Geometry of Numbers*. Springer, Berlin. Second printing, 1971.

[2] Edmonds, J. (1967). Systems of Distinct Representatives and Linear Algebra. *J. Res. Nat. Bur. Standards Sect. B* **71B** 241–245.

[3] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability, A Guide to the Theory of NP-Completeness*. Freeman & Co., San Francisco.

[4] Grotschel, M., Lovász, L. and Schrijver, A. (1981). The Ellipsoid Method and Its Consequences in Combinatorial Optimization *Combinatorica* **1** 169–197.

[5] Hirschberg, D. S. and Wong, C. K. (1976). A Polynomial-time Algorithm for the Knapsack Problem with Two Variables. *J. Assoc. Comput. Mach.* **23** 147–154

[6] Kannan, R. (1980). A Polynomial Algorithm for the Two-variable Integer Programming Problem *J. Assoc. Comput Mach* **27** 118–122

[7] ——— and Bachem, A. (1979) Polynomial Algorithms for Computing the Smith and Hermite Normal Forms of an Integer Matrix. *SIAM J. Comput.* **8** 499–507

[8] Khachiyan, L. G (1979). A Polynomial Algorithm in Linear Programming *Dokl. Akad Nauk SSSR* **244** 1093–1096 (English translation: *Soviet Math Dokl.* **20** (1979), 191–194)

[9] Lenstra, A. K., Lenstra, H. W., Jr. and Lovász, L (1982) Factoring Polynomials with Rational Coefficients. *Math. Ann.* **261** 515–534.

[10] Scarf, H. E. (1981). Production Sets with Indivisibilities—Part I Generalities *Econometrica* **49** 1–32 Part II The Case of Two Activities, ibid , 395–423.

[11] Van der Waerden, B. L. (1956) Die Reduktionstheorie von positiven quadratischen Formen *Acta Math* **96** 265–309.

[12] Von zur Gathen, J. and Sieveking, M. (1978). A Bound on Solutions of Linear Integer Equalities and Inequalities *Proc Amer. Math. Soc.* **72** 155–158

MATHEMATISCH INSTITUUT, UNIVERSITEIT VAN AMSTERDAM, ROETERSSTRAAT 15, 1018 WB AMSTERDAM, THE NETHERLANDS