On the Complexity of Integer Programming

CHRISTOS H. PAPADIMITRIOU

Massachusetts Institute of Technology, Cambridge, Massachusetts, and National Technical University, Athens, Greece

ABSTRACT. A simple proof that integer programming is in $\mathcal{NP}$ is given. The proof also establishes that there is a pseudopolynomial-time algorithm for integer programming with any (fixed) number of constraints.

KEY WORDS AND PHRASES: integer linear programming, $\mathcal{P}$, $\mathcal{NP}$, pseudopolynomial algorithms

CR CATEGORIES: 5 25, 5.3, 5.4

1. *Introduction*

The *knapsack problem* is the following one-line integer programming problem: Is there a 0–1 n-vector x such that

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b,$$

where $b, a_1, \ldots, a_n$ are given positive integers?

The knapsack problem is NP-complete [5, 7]. However, it is well known that it can be solved by a *pseudopolynomial* algorithm [4], that is, an algorithm with running time bounded by a polynomial in n *and* $a = \max\{a_1, \ldots, a_n, b\}$. Indeed, one can show quite easily that there is a pseudopolynomial-time algorithm for any one of the following extensions of the knapsack problem:

(a) The x_i are not restricted to be 0–1.
(b) Some of the a_i are negative.
(c) There are $m > 1$ equations to be satisfied (m fixed).

In fact, with a little care, pseudopolynomial algorithms can be developed for the combination of *any two* of these extensions. In this note we show that there is a pseudopolynomial algorithm for the problem that results by extending the knapsack problem in *all three* directions above.

Our proof settles another interesting question. It has been shown by many people (including [1, 2, 6]) that *integer programming* (i.e., the problem of deciding whether, for given $m \times n$ integer matrix A and m-vector b, the conditions

$$Ax = b, \qquad x \geq 0, \quad \text{integer},$$

are satisfied by some $x \in \mathbb{N}^n$) is in $\mathcal{NP}$. The proofs usually amount to showing

Author's address Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139.

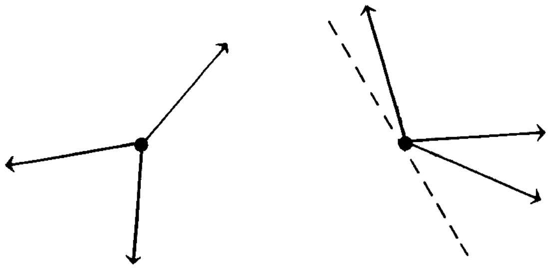

Figure 1

(a) (b)

that if the problem has a solution $x \in \mathbf{N}^n$, then it has another solution $x_0 \in \{0, 1, \ldots, a^{p(n)}\}^n$, where p is a polynomial and $a = \max_{i,j}\{|a_{ij}|, |b_j|\}$. We give here a considerably simpler proof of this fact. Furthermore, our bound is of the form $(an)^{p(m)}$. Since it is natural to assume that $m \le n$, this is a significant improvement.

In our proof we use several times the following simple lemma, easily proved from Cramer's rule.

LEMMA 1. *Let A be a nonsingular $m \times m$ integer matrix. Then the components of the solution of $Ax = b$ are all rationals with numerator and denominator bounded by $(ma)^m$, where $a = \max_{i,j} \{|a_{ij}|, |b_j|\}$.*

Our second lemma is a multidimensional, finite precision generalization of the following intuitive fact: If three directions on the plane cannot be left on the same side of any line through the origin (Figure 1), then they can be the directions of three balanced forces. It is a version of Farkas' Lemma.

LEMMA 2. *Let $v_1, \ldots, v_k$ be $k > 0$ vectors in $\{0, \pm 1, \pm 2, \ldots, \pm a\}^m$, and let $M = (ma)^{m+1}$. Then the following are equivalent:*

(*a*) *There exist k reals $\alpha_1, \ldots, \alpha_k \ge 0$, not all zero, such that $\sum_{j=1}^k \alpha_j v_j = \mathbf{0}$.*
(*b*) *There exist k integers $\alpha_1, \ldots, \alpha_k$, $0 \le \alpha_j \le M$ for $j = 1, \ldots, k$, not all zero, such that $\sum_{j=1}^k \alpha_j v_j = \mathbf{0}$.*
(*c*) *There is no vector $h \in \mathbf{R}^m$ such that $p_j = h^{\mathrm{T}} v_j > 0$ for $j = 1, \ldots, k$.*
(*d*) *There is no vector $h \in \{0, \pm 1, \ldots, \pm M\}^m$ such that $h^{\mathrm{T}} v_j \ge 1$ for $j = 1, \ldots, k$.*

PROOF

(a) $\Rightarrow$ (b). Follows from Lemma 1.

(b) $\Rightarrow$ (c). Suppose that such an h exists. Then $\mathbf{0} = h^{\mathrm{T}} \sum_{j=1}^k \alpha_j v_j = \sum_{j=1}^k \alpha_j p_j > 0$, absurd.

(c) $\Rightarrow$ (d). Trivial.

(d) $\Rightarrow$ (a). Using Lemma 1, it is easy to see that (d) is equivalent to saying that the linear program

$$\text{minimize } h^{\mathrm{T}} \cdot \mathbf{0} \qquad \text{subject to} \quad h^{\mathrm{T}} v_j = 1, \quad j = 1, \ldots, k,$$

is infeasible. Consequently, the *dual* linear program (see [3, 8])

$$\text{maximize} \quad \sum_{j=1}^{k} \alpha_j$$

$$\text{subject to} \quad \sum_{j=1}^{k} \alpha_j v_j = \mathbf{0} \quad \text{and} \quad \alpha_j \ge 0, \quad j = 1, \ldots, k,$$

is unbounded (because it *is* feasible, with $\alpha_j = 0$, all j) and therefore has a strictly positive solution. (a) follows. □

We are now ready to prove our main result.

THEOREM. *Let A be an $m \times n$ integer matrix and b an m-vector, both with entries from $\{0, \pm 1, \ldots, \pm a\}$. Then if $Ax = b$ has a solution $x \in \mathbb{N}^n$, it also has one in $\{0, 1, \ldots, n(ma)^{2m+1}\}^n$.*

PROOF. Let $M = (ma)^m$, and consider the smallest (say, w.r.t. sum of components) integer solution x to $Ax = b$. If all components of x are smaller than M, we are done. Otherwise assume, without loss of generality, that $x_j \geq M$ for $j = 1, \ldots, k$. Consider the first k columns of A, namely, $v_1, \ldots, v_k$.

Case 1. There exist integers $\alpha_1, \ldots, \alpha_k$ between 0 and M, not all zero, such that $\sum_{j=1}^{k} \alpha_j v_j = \mathbf{0}$. It follows that the vector $x' = (x_1 - \alpha_1, \ldots, x_k - \alpha_k, x_{k+1}, \ldots, x_n)$ is also a solution to $Ax = b$, thus contradicting the minimality of x.

Case 2. Not so. By Lemma 2 there is a vector $h \in \{0, \pm 1, \ldots, \pm M\}^m$ such that $h^{\mathrm{T}} v_j \geq 1$ for $j = 1, \ldots, k$. Let us premultiply the equation $Ax = b$ by h^{T}. We obtain $\sum_{j=1}^{k} h^{\mathrm{T}} v_j x_j = h^{\mathrm{T}} b - \sum_{j=k+1}^{n} h^{\mathrm{T}} v_j x_j$, and therefore $\sum_{j=1}^{k} x_j \leq n^2 maM^2 = n(ma)^{2m+1}$. The theorem follows. □

COROLLARY 1. *There is a pseudopolynomial algorithm for solving $m \times n$ integer programs, with fixed m.*

PROOF. We can solve the $m \times n$ integer program $Ax = b$ by dynamic programming, proceeding in stages. At the jth stage we compute the set S_j of all vectors v that can be written as $v = \sum_{i=1}^{j} v_i x_i$, with v_i the ith column of A and with the x_i in the range $0 \leq x_i \leq B$, where $B = n(ma)^{2m+1}$. Since the S_j cannot become larger than $(nB)^m$, the whole algorithm can be carried out in time $O((nB)^{m+1}) = O(n^{2m+2}(ma)^{(m+1)(2m+1)})$, a polynomial in n and a if m is fixed. □

We can extend Corollary 1 to the *optimization* version of integer programming, that is, the problem of finding the x which

$$\begin{aligned} &\text{minimizes} \quad c'x \\ &\text{subject to} \quad Ax = b, \quad x \geq 0, \quad \text{integer}. \end{aligned} \tag{1}$$

We first need the following lemma.

LEMMA 3. *Consider* (1) *and the following linear programming relaxation*:

$$\begin{aligned} &\textit{minimize} \quad c'x \\ &\textit{subject to} \quad Ax = b, \quad x \geq 0. \end{aligned} \tag{2}$$

If (1) *is feasible and* (2) *is unbounded, then* (1) *is also unbounded.*

PROOF. If (2) is unbounded, then it has a *feasible direction* y such that (a) the components of y are rationals, (b) $c'y < 0$, and (c) if x is feasible then $x + \lambda y$ is feasible for every $\lambda \geq 0$. For every feasible solution $x \in \mathbb{N}^n$ of (1), therefore, there is a set of other integer solutions of the form $x_j = x + jPy$, where $j \in \mathbb{N}$ and P is the product of the denominators of y. This set is of unbounded cost. □

LEMMA 4. *Suppose that* (1) *is feasible and bounded, and let z be its optimal cost. Then $|z| \leq (\sum_{j=1}^{n} |c_j|) \cdot M$, where $M = n^2(ma^2)^{2m+3}$.*

PROOF. That $z \leq (\sum_{j=1}^{n} |c_j|) \cdot M$ follows from the theorem. For a lower bound it is obvious that $z_2 \leq z$, where z_2 is the optimum cost of (2); notice that by Lemma 3,

(2) is bounded, given that (1) is. It is immediate, however, that $|z_2| \leq (\sum_{j=1}^{n} |c_j|)\cdot M$, because z_2 is attained at a basic feasible solution of (1). □

We therefore have

COROLLARY 2. *There is a pseudopolynomial algorithm for finding the optimum in any $m \times n$ optimization integer program* (1), *for m fixed.*

PROOF. We may simply solve one feasibility integer program,

$$\begin{aligned} c'x &= z, \\ Ax &= b, \\ x &\geq 0, \quad \text{integer}, \end{aligned}$$

for each value of z in the range

$$-\sum_{j=1}^{n} |c_j|\cdot M - 1 \leq z \leq \sum_{j=1}^{n} |c_j|\cdot M,$$

using the pseudopolynomial algorithm of Corollary 1. Binary search would yield a better bound. □

Notice that no pseudopolynomial algorithm is likely to exist for the general (not fixed m) integer programming problem, since this problem is *strongly* NP-complete (see [4]).

Note added in proof. Recently H. W. Lenstra discovered a remarkable polynomial-time algorithm for testing whether a system of linear *inequalities* in n variables has an integer feasible point, where n is fixed. Our result (Corollary 1), however, is independent of Lenstra's, since it is concerned with a fixed number of *equalities.* This problem is a generalization of the knapsack problem, and thus it is very unlikely that it can be solved by a polynomial-time algorithm.

REFERENCES

1 BOROSH, I, AND TREYBIG, L B. Bounds on positive integral solutions to linear Diophantine equations. *Proc Amer Math Soc 55* (1976), 299–304
2 COOK, S A Private communication, 1978
3 DANTZIG, G B *Linear Programming and Extensions* Princeton University Press, Princeton, N J, 1962
4 GAREY, M.R, AND JOHNSON, D S "Strong" NP-completeness results motivation, examples, and implications *J. ACM 25*, 3 (July 1978), 499–508.
5 GAREY, M.R, AND JOHNSON, D S *Computers and Intractability A Guide to the Theory of NP-completeness* Freeman, San Francisco, 1979
6 KANNAN, R, AND MONMA, C.L. On the computational complexity of integer programming problems In *Lecture Notes in Economics and Mathematical Systems, Vol 157*, Springer-Verlag, 1978, pp 161–172
7. KARP, R M. Reducibility among combinatorial problems In *Complexity of Computer Computations*, R E. Miller and J W Thatcher, Eds, Plenum, New York, 1972, pp 85–103.
8. PAPADIMITRIOU, C H, AND STEIGLITZ, K *Combinatorial Optimization Algorithms and Complexity* Prentice-Hall, Englewood Cliffs, N J, 1981

RECEIVED JANUARY 1980; REVISED JUNE 1980, ACCEPTED JUNE 1980