MIT/LCS/TM-152

ON THE COMPLEXITY OF INTEGER PROGRAMMING

Christos H. Papadimitriou

February 1980

On the Complexity of Integer Programming

Christos H. Papadimitriou
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Mass. 02139

January 1980

ABSTRACT

We give a simple proof that integer programming is in *NP*. Our proof also establishes that there is a pseudopolynomial time algorithm for integer programming with any (fixed) number of constraints.

Key Words: Integer Linear programming, *P*, and *NP*, pseudopolynomial algorithms.

The <u>knapsack problem</u> is the following one-line integer programming problem: Is there a 0-1 n-vector x such that

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

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

The knapsack problem is NP-complete ([Ka],[GJ2]). However, it is well-known that it can be solved by a <u>pseudopolynomial</u> algorithm [GJ1], that is, an algorithm with running time bounded by a polynomial in n <u>and</u> $a=\max\{a_1,\dots,a_n,b\}$. 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's are not restricted to be 0-1.

(b) Some of the a_i's 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 <u>any two</u> of these extensions. In this note we show that there is a pseudopolynomial algorithm for the problem resulting by extending the knapsack problem in <u>all three</u> directions above.

Our proof settles another interesting question. It has been shown by many people, including [BT], [KM] and [Co], that <u>integer programming</u> (that is, the problem of deciding whether, for given mxn integer matrix A and m-vector b, the conditions

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

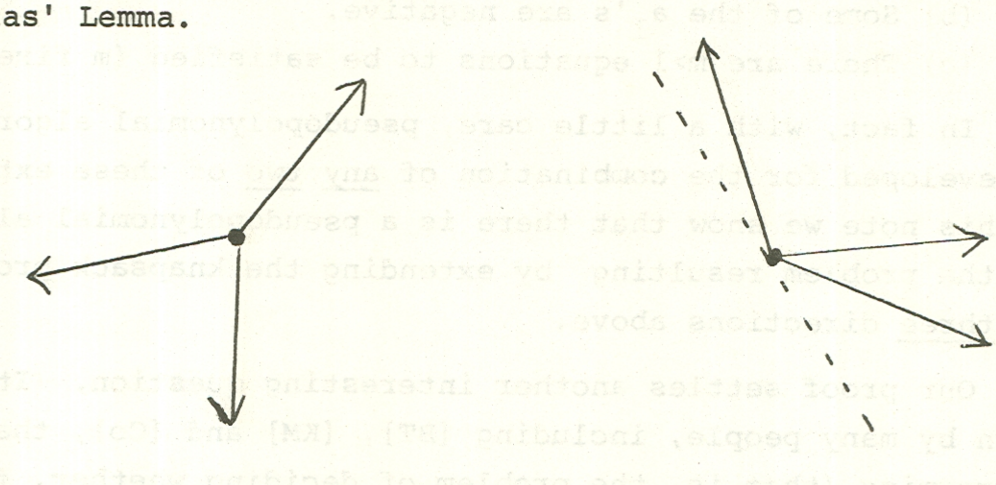
are satisfied by some $x\in\mathbb{N}^n$) is in *NP*. The proofs usually amount to showing that if the problem has a solution $x\in\mathbb{N}^n$, then it has another solution $x_0\in\{0,1,\dots,a^{p(n)}\}^n$, where p is a polynomial and $a = \max_{i,j}\{|a_{ij}|,|b_j|\}$. We give here a considerably simpler proof of this fact. Furthermore, our bound is of the form $(an)^{p(m)}$.

Since it is natural to assume that $m \leq n$, this is a significant improvement.

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

<u>Lemma 1</u> 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 $(ma)^{m+1}$, where $a = \max_{i,j}\{|a_{ij}|,|b_j|\}$.□

Our second Lemma is a multi-dimensional, finite precision generalization of the following intuitive fact: If three directions on the plane cannot be left in 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.



(a) (b)

Figure 1

<u>Lemma 2</u> Let $v_1 ..., v_k$ be $k>0$ vectors in $\{0,\pm 1,\pm 2,...,\pm a\}^m$, and let $M = (ma)^{m+1}$.

Then the following are equivalent:

(a) There exist k reals $\alpha_1, \ldots, \alpha_k \geq 0$, not all zero, such that $\sum_{j=1}^{k} \alpha_j v_j = \underline{0}$.

(b) There exist k integers $\alpha_1, \ldots, \alpha_k$ $0 \leq \alpha_j \leq M$ for $j=1\ldots,k$, not all zero, such that $\sum_{j=1}^{k} \alpha_j v_j = \underline{0}$.

(c) There is no vector $h \in \mathbb{R}^m$ such that $p_j = h^T v_j > 0$ for $j=1\ldots,k$.

(d) There is no vector $h \in \{0, \pm 1, \ldots, \pm M\}^m$ such that $h^T v_j \geq 1$ for $j=1,\ldots,k$.

Proof (a)⇒(b) Follows from Lemma 1.

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

(c)⇒(d) Trivial

(d)⇒(a) Using Lemma 1, it is easy to see that (d) is equivalent to saying that the linear program

$$\text{minimize } h^T \cdot \underline{0}, \text{ subject to } h^T v_j \geq 1, j=1,\ldots,k$$

is infeasible. Consequently, the dual linear program (see [Da],[PS])

$$\text{maximize } \sum_{j=1}^{k} \alpha_j, \text{ subject to } \sum_{j=1}^{k} \alpha_j v_j = \underline{0}, \text{ and } \alpha_j \geq 0, j=1,\ldots,k$$

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

We are now ready to prove our main result.

Theorem Let A be an m×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 $\{0,1,\ldots,n^2(ma^2)^{2m+3}\}^n$.

Proof Let $M = (ma)^{m+1}$, and consider the smallest (say, wrt sum of components) integer solution x to Ax=b. If all components of x are smaller than M, we are done. Otherwise assume that, without loss of generality, $x_j \geq M$ 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 and not all zero, such that $\sum_{j=1}^{k} \alpha_j v_j = \underline{0}$. It follows that also the vector $x' = (x_1-\alpha_1,\ldots,x_k-\alpha_k,x_{k+1},\ldots,x_n)$ is a solution to $Ax=0$, thus contradicting the minimality of x.

Case 2 Not so. By Lemma 2, there is a vector $h\in\{0,\pm1,\ldots,\pm M\}^m$ such that $h^T v_j \geq 1$ for $j=1,\ldots,k$. Let us premultiply the equation $Ax=b$ by h^T. We obtain $\sum_{j=1}^{k} h^T v_j x_j = h^T b - \sum_{j=k+1}^{n} h^T v_j x_j$, and therefore $\sum_{j=1}^{k} x_j \leq n^2 maM^2 = n^2 (ma)^{2m+3}$. The Theorem follows.□

Corrollary 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, prodeeding in stages. At the j^{th} 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 i^{th} column of A and with x_i's in the range $0\leq x_i \leq B$, where $B=n^2(ma)^{2m+3}$. Since the S_j's cannot become larger than $(nB)^m$, the whole algorithm can be carried out in time $O((nB)^{m+1}) = O(n^{3m+3}(ma)^{(m+1)(2m+3)})$, 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

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

We first need the following Lemma

Lemma 3 Consider (1) and the following linear programming relaxation.

$$\begin{aligned} &\text{minimize} && c'x \\ &\text{subject to} && Ax=b \qquad (2) \\ &&& x\geq 0. \end{aligned}$$

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 $x\in N^n$ 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 bounded, and let z be its optimal cost. Then $|z| \le (\sum_{j=1}^{n} |c_j|)\cdot M$, where $M = n^2(ma^2)^{2m+3}$.

Proof That $z \le (\sum_{j=1}^{n} |c_j|)\cdot M$ follows from the Theorem. For a lower bound, it is obvious that $z_2 \le z$, 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 $|z_2| \le (\sum_{j=1}^{n} |c_j|)\cdot M$.□

We therefore have

Corollary 2 There is a pseudopolynomial algorithm for finding the optimum in any m×n optimization integer program (1), for m fixed.

Proof We may simply solve one feasibility integer program

$$c'x = z$$
$$Ax = b$$
$$x \ge 0, \text{ integer}$$

for each value of z in the range

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

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 [GJ1]).

1. References

[BT] I. Borosh, L. B. Treybig, "Bounds on positive integral solutions to linear Diophantine equations" Proc. AMS 55, pp. 299-304, 1976.

[Co] S.A. Cook, private communication, 1978.

[Da] G.B. Dantzig, Linear Programming and Extensions, Princeton Univ. Press, Princeton, N.J., 1962.

[GJ1] M.R. Garey, D.S. Johnson, "Strong NP-completeness results: motivation, examples, and implications", JACM 25, pp. 499-508, 1978.

[GJ2] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide in the Theory of NP-completeness, Freeman, 1979.

[KM] R. Kannan, C.L. Monma, "On the computational complexity of integer programming problems". Report 7780-OR, Institut für Okonometrie und Operations Research, Bonn, 1977.

[Ka] R.M. Karp, "Reducibility among combinatorial problems" in R.E. Miller and J.W. Thatcher (eds.) Complexity of Computer Computations, Plenum, 1972.

[PS] C.H. Papadimitriou, K. Steiglitz Combinatorial Optimization: Algortihms and Complexity, Prentice-Hall, to appear, 1980.