SOLVING LINEAR PROGRAMMING PROBLEMS IN INTEGERS[1]

BY

RALPH E. GOMORY

The problem of finding the best solution in integers to a linear programming problem arises naturally in several ways. In an allocation of resources problem involving indivisible items such as ships, a solution involving fractions is one that cannot be realized in practice. Sometimes this matters and sometimes not. If the numbers involved are large and especially if the data is as poor as it sometimes can be in practical problems, one can round off to the nearest integer and probably not make too great an error. However, when a combinatorial problem is formulated as a linear programming problem, as in Dantzig [1], the data is usually quite precise and the numbers in the solution are often restricted to be zero or one. An example would be the task of selecting the largest possible expedition from a group of available people, subject to certain restrictions such as "persons 7 and 8 cannot both be included in the expedition." This is converted into a linear programming problem by assigning variables x_i to each of the people. In a solution $x_i = 0$ means that a person is included in the expedition, $x_i = 1$ means he is excluded. The problem then is to minimize $\sum_i x_i$ subject to a series of restrictions such as

$$x_7 + x_8 \geqq 1.$$

If a solution can be obtained in which the variables are 0 or 1, the inequality above implies that either person 7 or 8 has been excluded from the expedition. In a problem such as this one the numbers are automatically small (0 or 1) and a fractional solution is meaningless. In this particular example it is enough to demand a solution in integers. The minimization condition then assures that the solution contains only zeros and ones.

A close connection also exists between integer programming problems and problems involving piecewise linear, but not convex, domains or objective functions.

This paper outlines a finite algorithm for obtaining integer solutions to linear programs. The algorithm has been programmed successfully on an E101 computer and used to run off the integer solution to small (seven or less variables) linear programs completely automatically.[2]

---

[1] This work was supported in part by the Princeton–IBM mathematics research project.

[2] More recently a FORTRAN program on an IBM 704 has been used to run problems up to $m = n = 15$. The problems (only a few were run) ran rapidly.

The algorithm closely resembles the procedures already used by Dantzig, Fulkerson and Johnson [2], and Markowitz and Manne [3], to obtain solutions to discrete variable programming problems. Their procedure is essentially this. Given the linear program, first maximize the objective function using the simplex method, then examine the solution. If the solution is not in integers, ingenuity is used to formulate a new constraint that can be shown to be satisfied by the still unknown integer solution but not by the non-integer solution already attained. This additional constraint is added to the original ones, the solution already attained becomes non-feasible, and a new maximum satisfying the new constraint is sought. This process is repeated until an integer maximum is obtained, or until some argument shows that a nearby integer point is optimal.

What has been needed to transform this procedure into an algorithm is a systematic method for generating the new constraints. A proof that the method will actually give the integer solution in a finite number of steps is also important. This paper will describe an automatic method of generating new constraints. The proof of the finiteness of the process will be given separately.

Let us suppose that the original inequalities of the linear program have been replaced by equalities in nonnegative variables, so that the problem is to find nonnegative integers, $z, x_1, \cdots, x_m, t_1, \cdots, t_n$, satisfying

$$\begin{aligned} z &= a_{0,0} + a_{0,1}(-t_1)\cdots a_{0,n}(-t_n) \\ x_1 &= a_{1,0} + a_{1,1}(-t_1)\cdots a_{1,n}(-t_n) \\ &\vdots \\ x_m &= a_{m,0} + a_{m,1}(-t_1)\cdots a_{m,n}(-t_n) \end{aligned} \tag{1}$$

such that z is maximal. Using the method of pivot choice given by the simplex (or dual simplex) method, successive pivots result in leading the above array into the standard simplex form,

$$\begin{aligned} z &= a'_{0,0} + a'_{0,1}(-t'_1)\cdots a'_{0,n}(-t'_n) \\ x'_1 &= a'_{1,0} + \cdots \qquad\quad \cdots a'_{1,n}(-t'_n) \\ &\vdots \\ x'_m &= a'_{m,0} + \cdots \qquad\quad \cdots a'_{m,n}(-t'_n) \end{aligned} \tag{2}$$

where the primed variables are a rearrangement of the original variables and the $a'_{0,j}$ and $a'_{i,0}$ are nonnegative. From this array the simplex solution $t'_j = 0$, $x'_i = a'_{i,0}$ is read out.

An additional constraint can now be formulated. The constraint which will be generated is not unique, but is one of a large class that can be produced by a more systematic version of the following procedure.

Let us consider the equivalence relation, equivalence modulo 1.

We will write $a \equiv b$ (a equivalent to b) if and only if $a - b$ is an integer. This equivalence relation will be used to produce a new constraint.

If the $a'_{i,0}$ are not all integers, select some i_0 with $a'_{i_0,0}$ non-integer. For this i_0 we have the equation

$$x'_{i_0} = a'_{i_0,0} + \sum_{j=1}^{j=n} a'_{i_0,j}(-t'_j). \tag{3}$$

Any nonnegative *integer* solution $z'', x'', \cdots, x''_m, t''_1, \cdots, t''_n$ must satisfy (3). Since x''_{i_0} is an integer we have

$$x''_{i_0} \equiv 0. \tag{4}$$

So from (3) and (4) we have

$$\sum_{j=1}^{j=n} a'_{i_0,j} t''_j \equiv a'_{i_0,0}. \tag{5}$$

If $\bar{a}_{i_0,j}$ is any number equivalent to $a'_{i_0,j}$, then since the t''_j are integers,

$$\sum_{j=1}^{j=n} \bar{a}_{i_0,j} t''_j \equiv a'_{i_0,0}. \tag{6}$$

If the $\bar{a}_{i_0,j}$ chosen are all nonnegative, then the left hand side of (6) is also nonnegative since the t''_j are nonnegative by assumption. So the left hand side of (6) is both nonnegative and equivalent to $a'_{i_0,0}$. This implies

$$\sum_{j=1}^{j=n} \bar{a}_{i_0,j} t''_j \geqq f'_{i_0,0} \tag{7}$$

where $f'_{i_0,0}$ is the fractional part of $a'_{i_0,0}$.

This inequality, although satisfied by any nonnegative *integer* solution to (2) is not satisfied by the present simplex solution since the simplex solution has $t'_j = 0, j = 1, n$.

The only restrictions placed on the $\bar{a}_{i_0,j}$ so far is that they should be nonnegative and equivalent to the $a'_{i_0,j}$. If any one of the chosen $\bar{a}_{i_0,j}$ is replaced by a smaller equivalent number which is still nonnegative, the result is a new inequality which is easily seen to imply the old one. A succession of such replacements then results in a series of increasingly strong inequalities. The strongest possible one, which implies all the others is

$$\sum_{j=1}^{j=n} f'_{i_0,j} t''_j \geqq f'_{i_0,0} \tag{8}$$

where $f'_{i_0,j}$ are the fractional parts of the $a'_{i_0,j}$. That is $f'_{i_0,j} = a'_{i_0,j} - n'_{i_0,j}$ where $n'_{i_0,j}$ is the largest integer $\leqq a'_{i_0,j}$.

To transform (8) into an equation we introduce the variable s_1, required to be nonnegative, by

$$s_1 = -f'_{i_0,0} - \sum_{j=1}^{j=n} f'_{i_0,j}(-t'_j) \tag{9}$$

and add equation (9) to the set (2). The new set will be referred to as (2*). Since s_1 is the difference between the left and right sides of (8), and these

Example

Maximize $w = 2x + 3y + z$

$$8x - 8y \leqq 7$$

$$-x + 6y \leqq 9$$

$$x + y + z \leqq 6$$

Introduce slacks t_1, t_2, t_3

	1	$-x$	$-y$	$-z$
$w =$	0	-2	-3	-1
$t_1 =$	7	8	-8	0
$t_2 =$	9	-1	6^*	0
$t_3 =$	6	1	1	1

	1	$-x$	$-t_2$	$-z$
$w =$	$4\frac{1}{2}$	$-2\frac{1}{2}$	$\frac{1}{2}$	-1
$t_1 =$	19	$6\frac{2^*}{3}$	$1\frac{1}{3}$	0
$y =$	$1\frac{1}{2}$	$-\frac{1}{6}$	$\frac{1}{6}$	0
$t_3 =$	$4\frac{1}{2}$	$1\frac{1}{6}$	$-\frac{1}{6}$	1

	1	$-t_1$	$-t_2$	$-z$
$w =$	$11\frac{25}{40}$	$\frac{15}{40}$	1	-1
$x =$	$2\frac{34}{40}$	$\frac{6}{40}$	$\frac{8}{40}$	0
$y =$	$1\frac{39}{40}$	$\frac{1}{40}$	$\frac{8}{40}$	0
$t_3 =$	$1\frac{7}{40}$	$-\frac{7}{40}$	$-\frac{16}{40}$	1^*

	$+1$	$-t_1$	$-t_2$	$-t_3$
$w =$	$12\frac{32}{40}$	$\frac{8}{40}$	$\frac{24}{40}$	1
$x =$	$2\frac{34}{40}$	$\frac{6}{40}$	$\frac{8}{40}$	0
$y =$	$1\frac{39}{40}$	$\frac{1}{40}$	$\frac{8}{40}$	0
$z =$	$1\frac{7}{40}$	$-\frac{7}{40}$	$-\frac{16}{40}$	1
$s_1 =$	$-\frac{39}{40}$	$-\frac{1}{40}$	$-\frac{8^*}{40}$	0

End of regular simplex method

	$+1$	$-t_1$	$-s_1$	$-t_3$
$w =$	$9\frac{7}{8}$	$\frac{1}{8}$	3	1
$x =$	$1\frac{7}{8}$	$\frac{1}{8}$	1	0
$y =$	1	0	1	0
$z =$	$3\frac{1}{8}$	$-\frac{1}{8}$	2	1
$t_2 =$	$4\frac{7}{8}$	$\frac{1}{8}$	-5	0
$s_2 =$	$-\frac{7}{8}$	$-\frac{1^*}{8}$	0	0

	$+1$	$-s_2$	$-s_1$	$-t_3$
$w =$	$\underline{9}$	1	3	1
$x =$	$\underline{1}$	1	1	0
$y =$	$\underline{1}$	0	1	0
$z =$	$\underline{4}$	-1	2	1
$t_2 =$	$\underline{4}$	1	-5	0
$t_1 =$	$\underline{7}$	-8	0	0

Integer solution

two sides are equivalent for any integer solution, s_1 will always be integer whenever the other variables are, so we still require all the variables appearing in (2*) to be integers.

The procedure now is to maximize z over the solutions to (2*). This is done using the dual simplex method because all the $a'_{0,j}$ and $a'_{i,0}$ are already nonnegative, and $-f'_{i_0,0}$ is the only negative entry in the zero column of the equations (2*). This fact usually makes remaximization quite rapid. The process is then repeated if the new simplex maximum is non-integer.

Of course the equations (2*) involve one more equation than the equations (2), and an equation is added after each remaximization. However, the total number need never exceed $m + n + 2$. For if an s-variable, added earlier in the computation reappears among the variables on the left hand side of the equations after some remaximization, the equation involving it can simply be dropped, as the only equations that must be satisfied by a solution are the original ones. This limits the total number of s-variables present at one time to $n + 1$ or less.

Of course even the process just described involves an element of choice, any of the rows i of (2) with $a'_{i,0}$ non-integer might be chosen to generate the new relation. Some choices are better than others. A good rule of thumb based on the idea of "cutting" as deeply as possible with the new relation, and borne out by limited computational experience, is to choose the row with the largest fractional part $f_{i,0}$ in the zero column.

This class of possible additional constraints is not limited to those produced by the method described here since it is easily seen that some simple operations on and between rows preserve the properties needed in the additional relations. These operations can be used to produce systematically a family of additional relations from which a particularly effective cut or cuts can be selected. A discussion of this class of possible additional constraints together with a rule of choice of row which can be shown to bring the process to an end in a finite number of steps—thus providing a finite algorithm—requires some space and will be given as part of a more complete treatment in another place.

A small example, illustrating the method, is on the preceding page.

References

1. George B. Dantzig, *Discrete-variable extremum problems*, Operations Res. vol. 5 no. 2 (1957).

2. G. Dantzig, R. Fulkerson, and S. Johnson, *Solution of a large-scale traveling salesman problem*, J. Operations Res. Soc. Amer. vol. 2 no. 4 (1954).

3. Harry M. Markowitz and Alan S. Manne, *On the solution of discrete programming problems*, Econometrica vol. 25 no. 1 (1957).

PRINCETON UNIVERSITY,
PRINCETON, NEW JERSEY