

An Automatic Method of Solving Discrete Programming Problems

Author(s): A. H. Land and A. G. Doig

Source: *Econometrica*, Vol. 28, No. 3 (Jul., 1960), pp. 497-520

Published by: The Econometric Society

Stable URL: <http://www.jstor.org/stable/1910129>

Accessed: 02-04-2018 01:37 UTC

REFERENCES

Linked references are available on JSTOR for this article:

http://www.jstor.org/stable/1910129?seq=1&cid=pdf-reference#references_tab_contents

You may need to log in to JSTOR to access the linked references.

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at <http://about.jstor.org/terms>



The Econometric Society is collaborating with JSTOR to digitize, preserve and extend access to *Econometrica*

AN AUTOMATIC METHOD OF SOLVING DISCRETE PROGRAMMING PROBLEMS

BY A. H. LAND AND A. G. DOIG

In the classical linear programming problem the behaviour of continuous, nonnegative variables subject to a system of linear inequalities is investigated. One possible generalization of this problem is to relax the continuity condition on the variables. This paper presents a simple numerical algorithm for the solution of programming problems in which some or all of the variables can take only discrete values. The algorithm requires no special techniques beyond those used in ordinary linear programming, and lends itself to automatic computing. Its use is illustrated on two numerical examples.

1. INTRODUCTION

THERE IS A growing literature [1, 3, 5, 6] about optimization problems which could be formulated as linear programming problems with additional constraints that some or all of the variables may take only integral values. This form of linear programming arises whenever there are indivisibilities. It is not meaningful, for instance, to schedule 3-7/10 flights between two cities, or to undertake only 1/4 of the necessary setting up operation for running a job through a machine shop. Yet it is basic to linear programming that the variables are free to take on any positive value,¹ and this sort of answer is very likely to turn up.

In some cases, notably those which can be expressed as transport problems, the linear programming solution will itself yield discrete values of the variables. In other cases the percentage change in the maximand² from common sense rounding of the variables is sufficiently small to be neglected. But there remain many problems where the discrete variable constraints are significant and costly.

Until recently there was no general automatic routine for solving such problems, as opposed to procedures for proving the optimality of conjectured solutions, and the work reported here is intended to fill the gap. About the time of its completion an alternative method was proposed by Gomory [5] and subsequently extended by Beale [1]. Gomory's method

¹ Or more generally, any value within a bounded interval.

² We shall speak throughout of maximisation, but of course an exactly analogous argument applies to minimisation.

is based on the systematic addition of new constraints which are satisfied by a discrete variable solution but not by a continuous variable solution. At present, the published results apply only to problems in which all the variables are discrete, but a generalisation to the mixed case (i.e., in which not all the variables are required to be discrete) is known to exist. The mixed problem has been solved by Beale using a method in which the discrete variables appear as the parameters of a subsidiary linear programme which is expressed entirely in terms of continuous variables; the parameters of this continuous problem are themselves required to satisfy a pure discrete problem for which Gomory's technique may be employed. The method described here applies also to the mixed problem and although we have in fact worked only on a desk computer, we have borne in mind throughout that the algorithm should be susceptible to programming on an electronic computer. It is not suggested that this method should supersede successful ad hoc methods for particular problems. It may, in fact, be chiefly useful for testing the validity of proposed ad hoc methods for new problems.

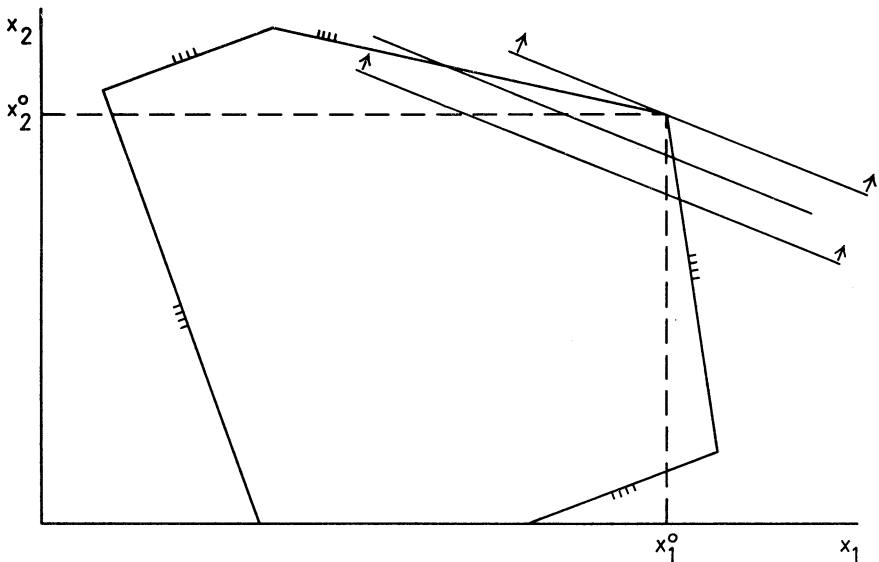


FIGURE 1

2. FORMULATION

The discrete programming problem can be expressed as follows:
Maximize

$$(1) \quad c'x + \bar{c}'y = \gamma,$$

subject to the constraints

$$(2) \quad Ax + \bar{A}y \leq b,$$

(3) x is a column vector with nonnegative integral components

(4) $y \geq 0$,

where γ , the maximand, is a scalar, b is a column vector of m rows, c and x are column vectors of n_1 rows, \bar{c} and y are column vectors of $(n - n_1)$ rows, A is a matrix of order $m \times n_1$, and \bar{A} is a matrix of order $m \times (n - n_1)$. A feasible solution of the problem is one which satisfies (2), (3), and (4).

A two variable linear programming problem without discrete variable constraints is illustrated in Figure 1 where it can be seen that the functional (represented by the family of parallel lines) reaches its maximum at (x_1^0, x_2^0) . The discrete variable constraints limit the set of feasible solutions to points within the original set for which both coordinates are integers; in Figure 2

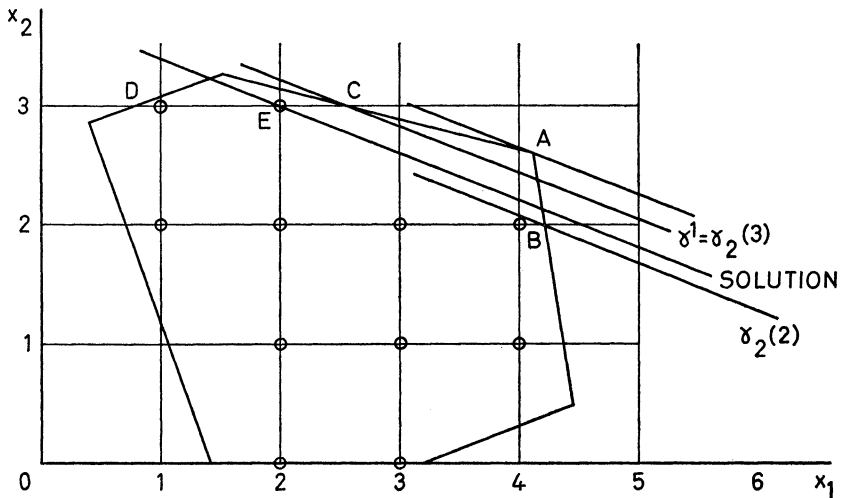


FIGURE 2

the complete set of feasible solutions to the discrete programming problem consists of the eleven points which have been circled. It is easy to see in this two dimensional case that the maximum solution is $x_1 = 2$, $x_2 = 3$. The procedure for arriving at this conclusion could be described as "pushing down the functional line until it meets an integral point." The algorithm to be described here is the generalisation of this procedure to many dimensions.

3. DESCRIPTION OF THE METHOD

As the set of feasible solutions to the discrete variable problem is not convex, any method which examines merely the immediate neighbourhood of a proposed solution can prove the existence only of a local optimum. The method used here makes systematic parallel shifts in the functional hyperplane in the direction of a reduction of the maximand, until a point

within the ordinary linear programming set is found which has integral coordinates in the specified (x variable) dimensions. This is obvious in principle but cannot be so readily done in practice as one does not have the faculty of "seeing" a hyperplane in n -dimensional space in order to determine if it contains a point whose x coordinates are all integers. Numerical methods can only examine one point at a time. Rules must therefore be devised to switch attention from one region of the falling functional hyperplane to another to ensure that no integer point has been passed.

In other words an upper bound to the functional γ^0 , is first obtained by solving the ordinary linear programming problem without the discrete variable constraints, and then successively more restrictive upper bounds, $\gamma^1, \gamma^2, \dots, \gamma^k$, are found. If the upper bound of the functional at any stage is γ^k , then it has been proved that there is no discrete variable solution with a higher value of the maximand than γ^k .

Consider the convex set of solutions of an ordinary linear programming problem, such as might be represented for a two-dimensional case by Figure 2. Any feasible value of the maximand, say γ^k , is uniquely associated with a definite position of the functional hyperplane which in general cuts through the n -dimensional convex set, and in the special case of the optimum value just touches the convex set. The intersection of the hyperplane with the

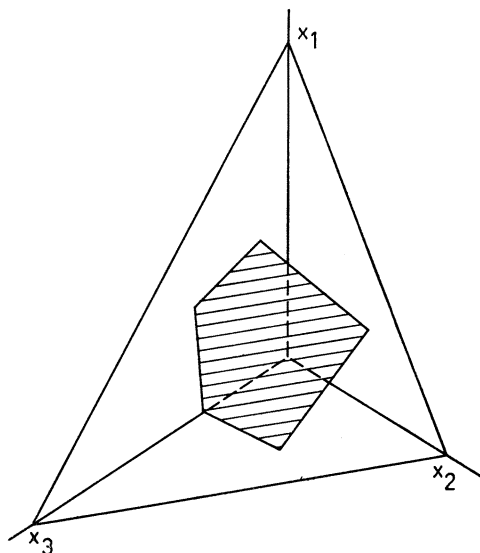


FIGURE 3

original convex set is itself a convex set of $n - 1$ dimensions. For example, Figure 3 represents such an intersection with a three-dimensional set.

The points x_1 , x_2 , and x_3 are the intercepts of the hyperplane on the 3 axes respectively and the shaded area represents the convex set which

lies within the constraints of the ordinary linear programming problem.

In such an $(n-1)$ -dimensional set there would be a minimum and a maximum value for each variable as shown in Figure 4.

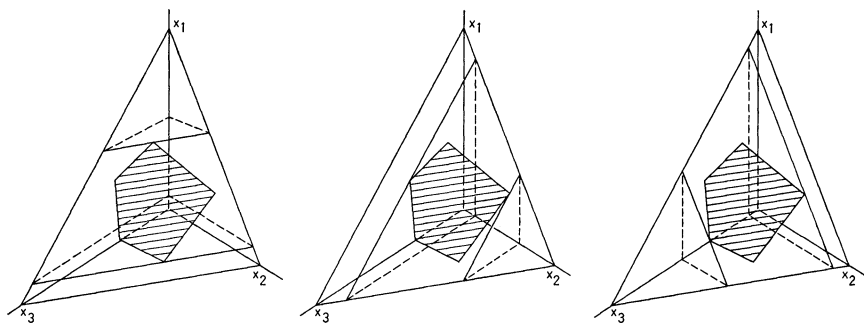


FIGURE 4

If $\gamma^k = \gamma^0$, i.e., if the functional hyperplane is that associated with the optimum solution of the ordinary linear programming problem, this convex set would normally consist of a single point (unless there were multiple optimum solutions). It is still true to say, however, that for every value of the maximand (position of the functional hyperplane) there is a minimum and a maximum value (which may coincide) for each x variable, consistent with the ordinary linear programming constraints.

Knowing for a particular value of γ the minimum and maximum value for each variable one knows also whether there is a possible integer value of each variable for that position of the hyperplane. If there is not at least one possible integer value for every x -variable, then one can say immediately that there is no solution to the problem at that value of the maximand. Unfortunately the converse is not true. The fact that each x variable takes an integer value at some point or points within the set is not a sufficient condition for there being an *intersection* of these integral coordinates within the set, and hence a feasible solution.

Since there is a unique minimum and maximum value for any variable x_k at a particular value of γ , it follows that one can define two functional relationships, $\min x_k$ and $\max x_k$, between x_k and the falling functional hyperplane (value of γ). The connection between these two functions and the fundamental problem may be demonstrated by means of the following system of inequalities:

$$\begin{array}{ll}
 (1') & c'x + \bar{c}'y - \gamma = 0, \\
 (2) & Ax + \bar{A}y \leq b, \\
 (3') & x \geq 0, \\
 (4) & y \geq 0, \\
 (5) & \gamma \begin{array}{l} \geq 0, \\ < 0, \end{array}
 \end{array}$$

This system defines a convex polyhedral set in $(n+1)$ -dimensional space and since the plane projection of a convex set is convex, the projection of this set onto the (x_k, γ) plane will yield a convex polygon whose upper (lower) boundary is a concave (convex) function of the abscissa. One such polygon is shown in Figure 5.

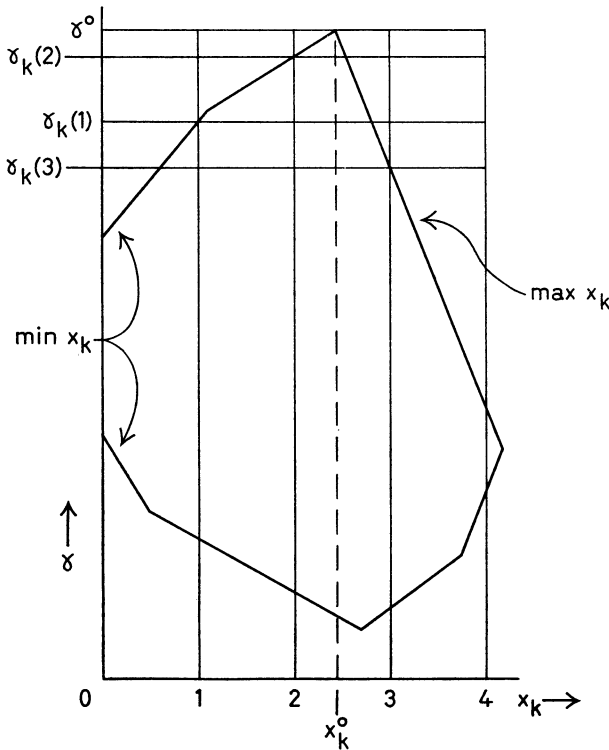


FIGURE 5

The value γ^0 which γ takes at the highest point of the polygon is the maximum value γ can take subject to the given inequality system. This is equivalent to saying that γ^0 is the optimal value of the functional γ in a linear programming problem with constraints (2), (3'), and (4). Further, x_k takes the value x_k^0 in this solution (if x_k were not in the optimum basis, the peak of the polygon would lie on the γ axis). The boundary of the polygon consists of the functions³ $\min x_k$ and $\max x_k$. These functions could be calculated by solving the two families of linear programming problems defined by (1'), (2), (3'), (4), and (5) with functionals "minimize x_k " and

³ In a minimization problem, we are interested in the *lower* boundary of the polygon; if all coefficients of A , \bar{A} and b are nonnegative, the point (x_k^0, γ^0) is at infinity and the polygon is unbounded; in this case, the functions $\min x_k$ and $\max x_k$ are monotonic.

“maximize x_k ,” for all possible values of γ using straightforward parametric linear programming [4, 7, 8]. A two-dimensional polygon of this type could be determined for every x variable of the original discrete problem.

In Figure 5, x_k^0 lies between the values 2 and 3. If $\min x_k$ and $\max x_k$ are traced out by reducing γ from γ^0 until an integer value of x_k is first encountered on each, the two values $\gamma_k(2)$ and $\gamma_k(3)$ on Figure 5 are obtained. This is equivalent to examining the path traced out by two specific corners of the intersection of the functional hyperplane with the convex set of feasible solutions when that hyperplane is systematically pushed back from its maximum position, γ^0 . It may be said of $\gamma_k(2)$ that it is an upper bound on the maximand since at no higher value of γ can x_k take an integer value.

The solution of the discrete programming problem will be developed by the systematic use of this argument. For convenience of exposition, we define a set of subsidiary problems, $P(j)$, as follows:

$P(j)$: Maximize γ subject to constraints (2), (3'), and (4) above, and the additional constraints that j of the x variables be nonnegative integers ($j = 0, 1, 2, \dots, n_1$).

Let S_j be the set of all feasible solutions to problems of type $P(j)$ and let \bar{S} be the set of nonfeasible solutions to any problem of type $P(j)$. The particular problem in which, for example, x_2 and x_4 are constrained to be nonnegative integers, will be written $P(2; 2, 4)$, and the set of its feasible solutions is written $S_2(2, 4)$. In this notation, the required solution is the element of S_{n_1} for which γ is maximized. The γ value of this solution is bounded above by the maximum value of γ over the set S_{n_1-1} which is itself bounded by the maximum value of γ over S_{n_1-2} , etc. The ultimate upper bound reached in this manner is γ^0 , the maximum value of γ over S_0 .

The solution will be constructed in the form of a tree graph whose vertices are elements either of one of the sets S_j , $j = 0, 1, 2, \dots, n_1$, or of the set \bar{S} . The steps of this construction are:

Step 0. The first vertex of the tree is the optimum solution of $P(0)$, which is now given the *label* γ^0 . If this solution also satisfies $P(n_1)$, it is the required solution.

Step 1. If γ^0 is not the required solution then, according to the rules given below, an arc is drawn to each of two points in S_1 (or, possibly, in \bar{S}). γ is evaluated at these points if they are in S_1 , but the γ value of a point in \bar{S} need never be calculated.

Step 2. If vertices $\gamma^0, \gamma^1, \gamma^2, \dots, \gamma^{k-1}$, have already been labelled, the

highest (according to the value of γ) unlabelled vertex not in \bar{S} is given the label γ^k .

Step 3. The final solution has been reached when for the first time a labelled vertex is an element of S_{n_1} ; this occurs as soon as all the x variables of a solution are nonnegative integers. If γ^k is not such a solution, suppose it to be an element of S_j . A new arc is drawn originating at the (labelled) vertex immediately above γ^k (this is not necessarily γ^{k-1}) and terminating at a point in the same subset of solutions (S_j) as γ^k or in \bar{S} . If this new vertex is in S_j , its γ value is necessarily less than or equal to γ^k .

Step 4. Two arcs are drawn from γ^k to points in S_{j+1} or in \bar{S} . Again, if these points are in S_{j+1} , their γ values are less than or equal to γ^k . The last two steps add three new arcs and vertices to the tree. Return to Step 2.

The labelled vertices form a sequence of nonincreasing upper bounds on the γ value of the final solution. In addition, the γ value of an unlabelled vertex of a set S_j cannot exceed that of a labelled vertex and therefore, at the time that it is labelled, γ^k is the greatest current upper bound on the optimal value of γ . Thus, provided the method used to add arcs and vertices to the graph covers all possibilities, the algorithm must lead to the optimum solution of $P(n_1)$, provided such a solution exists.

Rules for adding arcs and vertices. At Step 0, the tree consists of the single labelled vertex γ^0 which is an element of S_0 .

The decision is now made that x_r , which is in the basis at γ^0 , is to be constrained to nonnegative integral values, i.e., attention is focused on the problem $P(1; r)$. Let $x_r = x_r^0$ at γ^0 . The functions $\min x_r$ and $\max x_r$ are traced as far as the points $([x_r^0], \gamma_{rm})$ and $([x_r^0] + 1, \gamma_{rM})$ respectively, where:

$[x_r^0]$ is the greatest integer less than or equal to x_r^0 ,

γ_{rm} is the maximum value of γ subject to $P(0)$ and the condition

$x_r = [x_r^0]$, and

γ_{rM} is the maximum value of γ subject to $P(0)$ and the condition

$x_r = [x_r^0] + 1$.

The vertices γ_{rm} and γ_{rM} , if they exist, are elements of S_1 . It is seen, however, that both of them may be evaluated by solving a standard linear programming problem in $(n - 1)$ variables. If one of these problems, say "maximize γ , subject to $P(0)$ and to $x_r = [x_r^0]$ " is not feasible, γ_{rm} does not exist and a vertex is added to \bar{S} instead. This implies that S_1 does not contain an element for which $x_r = [x_r^0]$, which in turn implies that only values of x_r which satisfy $x_r \geq [x_r^0] + 1$ need be considered in developing the tree. If neither γ_{rm} nor γ_{rM} exist, x_r cannot be constrained to an integral

value and the problem possesses no feasible solution. We have now added the two arcs and vertices of Step 1.

From Step 2, $\gamma^1 = \max(\gamma_{rm}, \gamma_{rM})$. In calculating γ_{rm} and γ_{rM} , we have in effect examined every value of γ between γ^0 and γ^1 , and have shown that γ^1 is the maximum value of γ which is compatible with an integral value of x_r . Suppose $x_r = v$ at γ^1 . This equation is added to the problem as a new constraint which applies to any branch of the tree which can be traced back to γ^1 . The variable x_r is dropped from the calculations, thus reducing the dimensions of the set of solutions being explored to $(n - 1)$, and each boundary value b_i is reduced by an amount $a_{ir}v$. This new constraint is certainly valid with respect to solutions of $P(1; r)$ over the range of values of γ for which the only permissible integer value of x_r is v . The upper end of this range is γ^1 and, since the upper boundary of the polygon of Figure 5 is a concave function of the abscissa, the lower end must be one of $\gamma_r(v - 1)$ and $\gamma_r(v + 1)$. The larger of these two is the second best solution of $P(1; r)$.⁴ One of the neighbouring values is already known (it is $\min(\gamma_{rm}, \gamma_{rM})$) and an upper bound to the other can be found by extrapolating the appropriate function of x_r .⁵ If this upper bound exceeds the other neighbouring value, the corresponding γ value should be determined exactly. The arc and vertex of Step 3 have now been added to the tree.

To illustrate the argument, consider the two-dimensional problem shown in Figure 2.

Step 0. The solution at A provides the first vertex of the tree, γ^0 .

Step 1. The variable x_2 is selected and the points B ($x_2 = 2$, $\gamma = \gamma_{2m} = \gamma_2(2)$) and C ($x_2 = 3$, $\gamma = \gamma_{2M} = \gamma_2(3)$) are determined.

Step 2. $\gamma^1 = \gamma_{2M}$ for which $v = 3$.

Step 3. γ^1 is not the final solution. Since $x_2 = 4$ lies entirely outside the convex set S_0 , $\gamma_2(4)$ is in \bar{S} . The only possible integral value of x_2 for all values of γ satisfying $\gamma_2(2) < \gamma \leq \gamma_2(3)$ is $x_2 = 3$; this constraint is valid for the line segment CD which is a one-dimensional subset of the set of feasible solutions to $P(0)$.

⁴ Should it happen that, for example, $\gamma_r(v - 1) = \gamma^1$, $\gamma_r(v - 2)$ must be determined, and this step down of the integral argument of γ_r is continued until a value of $\gamma_r(v - w)$, ($1 \leq w \leq v$), is strictly less than γ^1 , or until a minimum integral value of x_r consistent with the ordinary linear programming constraints is reached. A parallel calculation will be needed for each integer value of x_r for which $\gamma_r(x_r) = \gamma^1$.

⁵ This procedure is used to calculate γ for $x_1 = 0$ in Appendix 1.

Step 4. x_1 is now constrained to integral values. One of the new vertices (γ_{1m}) is in S_2 and the other is in \bar{S} .

Step 2. $\gamma^2 = \gamma_{1m} > \gamma_{2M}$.

Step 3. γ^2 is an element of S_2 ; therefore the point E , corresponding to γ^2 is the required solution. At this point, $x_1 = 2, x_2 = 3$.

Clearly at any stage of the solution, the only x variables which may be violating the discrete constraints are those in the current basis. Therefore, by successively constraining each such variable in the manner described above, either a feasible solution of $P(n_1)$ will be found or else it will be shown that no such solution exists.

In the general case, the first upper bound on the functional, γ^0 , is the optimum solution to $P(0)$. The second upper bound is $\gamma^1 \leq \gamma^0$; γ^1 is the optimum solution to $P(1; r)$. This second upper bound could have been sharpened by finding the optimum solution to all problems of type $P(1)$ and selecting the *least* of these as γ^1 . Just as in the simplex method, however, it is not normally useful to determine which basis change yields the greatest increase in the functional, in this analysis it is usually more economical of computing effort to trace the min and max functions of one variable only at each stage. The criteria for selecting the appropriate variable are discussed in the appendices.

If γ^1 is not a solution to $P(n_1)$, a new x variable is chosen from those in the basis at γ^1 . Step 4 consists of tracing the max and min functions of this variable to the nearest integer values respectively above and below its value at γ^1 . This adds two new vertices to the tree, and, from Step 2, γ^2 is the maximum value of γ taken over these and the neighbouring values $\gamma_r(v-1)$ and $\gamma_r(v+1)$ of γ^1 . The whole argument can now be repeated with γ^2 replacing γ^1 as the current upper bound on the optimal value of γ . By continuing this process, a tree is formed each of whose vertices represents a known set of integer constraints (γ^1 , for example, represents $x_r = v$). A branch terminates if it reaches a vertex in \bar{S} . Ultimately, either all branches have terminated in \bar{S} and the problem has no feasible solution, or else a vertex, γ^f say, is reached for which all x variables are nonnegative integers. In principle, appropriate constraints are now applied to any x variable which has not yet been *constrained* at an integral value, and a vertex γ^F in S_{n_1} is reached which has the same γ value as the labelled vertex γ^f . Since the labelled vertices are upper bounds on the functional, γ^F must be the required optimal solution.

In practical terms, one would not necessarily stop a block of computations in the middle once the upper bound of γ with the current set of in-

tegral constraints had been proved to be lower than some value of γ which is still possible with a different set of constraints, since it may later be necessary to return to that branch. So long as a discrete solution when found is checked to determine that the upper bound of γ on every branch of the computation is at least as low as the solution value, the algorithm must yield the maximum solution. In automatic computation, where there is the problem of storing the bases associated with each branch of the tree, the best procedure may be to carry each branch in turn down to some predetermined "cut off" value of γ . Since in many discrete programming problems, it is easy to find a "good" as opposed to an optimum solution, there should be no difficulty in selecting a cut off value sufficiently low to ensure that the tree extends as far as a solution, and hence that the optimum solution has not been excluded from the tree. The cut off value can be raised during the course of the calculation as soon as better discrete solutions are found. Alternatively, if a high cut off value were chosen, the output of the computer could include sufficient information on each branch to enable the calculation to be restarted if no discrete solution were reached above that value.

4. NUMERICAL EXAMPLE

To illustrate the preceding sections, consider the following problem:⁶
Maximize

$$(1) \quad \gamma = c'x + \bar{c}'y$$

subject to

$$(2) \quad b = Px + Qy,$$

$$(3) \quad x \text{ to be a vector of nonnegative integers,}$$

$$(3') \quad x \geq 0,$$

$$(4) \quad y \geq 0,$$

where:

$$\begin{array}{l} x', y' = [\quad x_1 \quad x_2 \quad x_3 \quad | \quad y_1 \quad y_2 \quad y_3 \quad y_4 \quad] \\ P, Q = \left[\begin{array}{ccc|ccc} 10.9 & 3.6 & -40.8 & 43.9 & 7.1 & 1 & 0 \\ -86.8 & 32.7 & 24.3 & 13.8 & -12.6 & 0 & 1 \\ 60.9 & 68.9 & 69.0 & -56.9 & 22.5 & 0 & 0 \end{array} \right] b = \begin{bmatrix} 82.3 \\ 77.3 \\ 86.5 \end{bmatrix} \\ c', \bar{c}' = [\quad 77.9 \quad 76.8 \quad 89.6 \quad | \quad 97.1 \quad 31.3 \quad 0 \quad 0 \quad] \end{array}$$

⁶ Two possible computational routines, one based on the solution of many simple linear programmes and the other on parametric linear programming are given in Appendix 1.

The tree corresponding to the sequence of the calculations is given in Figure 6.

Step 0. The solution to the problem defined by (1), (2), (3') and (4) is

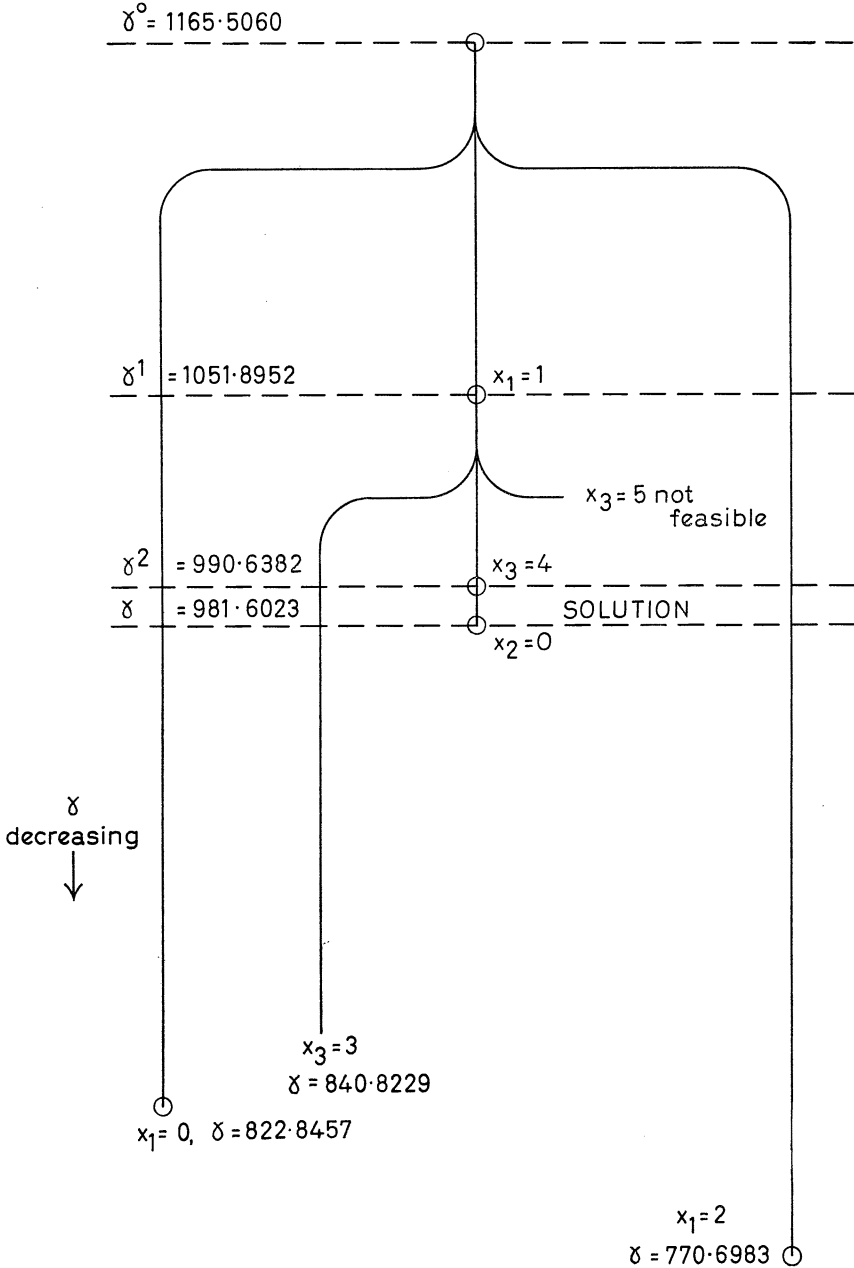


FIGURE 6

used as a starting point to the solution of the complete problem. In this solution

$$\gamma = \gamma^0 = 1165.5060.^7$$

Step 1. The upper bound of γ and the current solution at the start of this step are

$$\begin{aligned}\gamma^0 &= 1165.5060 \\ x_1 &= 1.4960 \\ x_3 &= 5.0210 \\ y_1 &= 6.1697.\end{aligned}$$

As x_1 is furthest from an integer value, find the first integer value on the two functions $\min x_1$ and $\max x_1$.

1.1. $\min x_1$. As γ decreases the first integer value of x_1 which is encountered is $x_1 = 1$ at $\gamma = 1051.8952$.

1.2. $\max x_1$. The first integer encountered is $x_1 = 2$ at $\gamma = 770.6983$.

Step 2. The greater of these two is $\gamma = 1051.8952$ at $x_1 = 1$, and this is given the label γ^1 . To complete this stage it is necessary to know the values of γ for the integers on either side of the one which is to be pursued.⁸ Therefore:

Step 3. $\min x_1$ is continued as far as its second integer value, $x_1 = 0$, at $\gamma = 822.8457$.

These three values for γ are recorded in a list of values of γ which will be pursued in their order of magnitude from the highest down. Thus, at the beginning of each Step 2, one value will be taken from the list and at the end of Steps 3 and 4, three (lower) values will have been added to it. The computation is complete when all γ values remaining on the list are lower than a γ value associated with an integer solution. At this stage, the list is:

γ		obtained in step
1051.8952	γ^1	1
770.6983		1
822.8457		3

⁷ For ease of presentation, the results have been rounded to four decimal places.

⁸ In a larger problem, where several branches have to be pursued, it may be necessary to continue the \min or \max functions further. E.g., if $x_1 = 2$, $\gamma = 770.6983$ had to be picked up and explored (there being no integer solution for $\gamma > 770.6983$), it would be necessary to compute γ for $x_1 = 3$.

The highest value is 1051.8952 which has therefore been labelled γ^1 (Step 2). The solution and constraints at this point are:

$$\begin{array}{ll} \text{Constraints:} & x_1 = 1. \\ \text{Solution:} & \gamma^1 = 1051.8952, \\ & x_1 = 1, \\ & x_3 = 4.4089, \\ & y_1 = 5.4838, \\ & y_2 = 1.4851. \end{array}$$

Step 4. 4.1. Min x_3 . At $x_3 = 4$, $\gamma = 990.6382$.

4.2. Max x_3 . The first integer encountered is $x_3 = 4$ at $\gamma = 803.4034$.⁹ The list of γ values is now:

γ	γ^1	obtained in step
1051.8952		1
770.6983		1
822.8457		3
990.6382		4

Step 2. The highest unlabelled value is 990.6382, which therefore becomes γ^2 .

Step 3. Carry min x_3 one step further. At $x_3 = 3$, $\gamma \leq 840.8229$.¹⁰ The solution and constraints are now:

$$\begin{array}{ll} \text{Constraints:} & x_1 = 1, x_3 = 4. \\ \text{Solution:} & \gamma^2 = 990.6382, \\ & x_1 = 1, \\ & x_2 = 0.2768, \\ & x_3 = 4, \\ & y_1 = 5.1513, \\ & y_2 = 1.0507. \end{array}$$

Step 4. 4.1. Min x_2 . At $x_2 = 0$, $\gamma = 981.6023$.

4.2. Max x_2 . This is a falling function. I.e., $x_2 = 1$ is not feasible, and another point is added to \bar{S} . Hence we have a solution at:

⁹ This apparently anomalous result arises because max x_3 is a falling function of γ . It implies that the point for which $x_1 = 1$, $x_3 = 5$ lies in \bar{S} since the problem with these constraints added is infeasible. It also implies that no feasible solution can be found in which $x_1 = 1$, $x_3 = 4$ and γ is less than 803.4034.

¹⁰ This is an upper bound to the branch value of γ which has been obtained by ignoring the fact that y_2 would be negative at this point. The true branch value cannot exceed 840.8229, but it will be determined exactly only if the branch becomes active.

$$\begin{aligned}
 \gamma &= 981.6023 \\
 x_1 &= 1 \\
 x_2 &= 0 \\
 x_3 &= 4 \\
 y_1 &= 5.0702 \\
 y_2 &= 1.6930 .
 \end{aligned}$$

As this value of γ is higher than any remaining on the list, this is the optimum solution.

APPENDIX 1

COMPUTATIONAL METHOD

The starting point for both methods described in this appendix is the solution of the linear programming problem described by (1), (2), (3') and (4) of Section 4, hereafter referred to as the initial problem. The integer restrictions (3) must now be considered. Since the two-dimensional set enclosed by $\min x_k$ and $\max x_k$ in the (x_k, γ) -plane is convex it follows that the first points for which γ must be determined are the integers on either side of the (nonintegral) value of one of the x variables. In principle, the routine will terminate irrespective of the x variable selected. But the limited experience afforded by the solution of two moderate size problems (Appendix 2) suggests that the criterion for selecting x_r has a considerable effect on the amount of computation needed. At first, x_r was chosen so as to maximize the decrease in γ at each step, which meant that the max and min functions had to be traced out for every x variable which was still nonintegral. As the calculations proceeded, it appeared that a more economical criterion (from the computational viewpoint) at each step would be to select the x variable which was furthest from an integer, and this rule was used in the remainder of the work. Another criterion which, however, has not been tested chooses the largest x variable with the object of making the sum of the constraints as large as possible. In some sense it must be true to say that the greater the sum of the constrained variables at any stage the more restricted is the remaining linear programming problem, and therefore the easier it is to solve.

In the example worked γ was determined for the points:

1. intersection of $\min x_1$ with $x_1 = 1$; and
2. intersection of $\max x_1$ with $x_1 = 2$.

Method 1. These two values of γ are found by solving the two linear programmes obtained by adding each of the above integral constraints in turn to the initial problem.

<i>Additional constraint</i>	γ
$x_1 = 1$	1051.8952
$x_1 = 2$	770.6983 .

The remaining points on the tree of Figure 6 can be found similarly. In all, the nine linear programmes which have to be solved are:

1. Constraints (2), (3') and (4) γ_0

2. Constraints (2), (3') and (4) and $x_1 = 1$ γ_1
3. Constraints (2), (3') and (4) and $x_1 = 2$
4. Constraints (2), (3') and (4) and $x_1 = 0$
5. Constraints (2), (3') and (4) and $x_1 = 1, x_3 = 4$ γ^2
6. Constraints (2), (3') and (4) and $x_1 = 1, x_3 = 5$ (not feasible) \bar{S}
7. Constraints (2), (3') and (4) and $x_1 = 1, x_3 = 3$
8. Constraints (2), (3') and (4) and $x_1 = 1, x_3 = 4, x_2 = 0$ SOLUTION
9. Constraints (2), (3') and (4) and $x_1 = 1, x_3 = 4, x_2 = 1$ (not feasible) \bar{S}

The result for programme 6 indicates that $\max x_3$ (with $x_1 = 1$) is a falling function of γ . The intersection of $\max x_3$ with $x_3 = 4$ lies on the lower boundary of the $(n-2)$ -dimensional set of feasible solutions for which $x_1 = 1, x_3 = 4$. It is represented by the point $(4, \gamma_u)$ in Figure 7. Since γ cannot be lowered below this unknown value γ_u ,

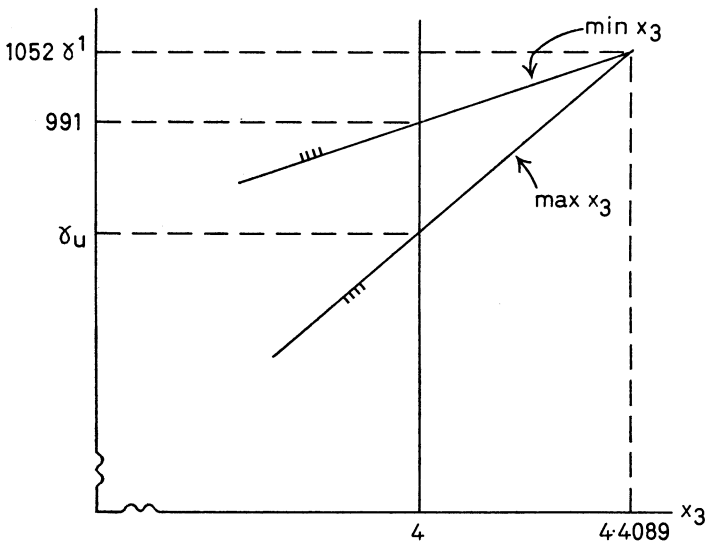


FIGURE 7

whilst still keeping x_3 fixed at 4, the point $x_1 = 1, x_3 = 4, \gamma = \gamma_u$ cannot be used as a starting point for exploration. Thus the fact that this method does not evaluate γ_u is irrelevant¹¹.

This method has the advantage that it uses only existing standard linear programming routines. If the time required to reach a solution is of paramount importance, e.g., for regular computer work or in desk computations, this method is slower than that based on parametric linear programming which is described in Method 2, below.

Method 2. The functional hyperplane is introduced as an additional (variable) constraint. This adds an extra row and column to the inverse basis, the entries in which can be easily deduced from the corresponding inverse basis in the initial problem. In the latter, let the first tableau be written in the form:

¹¹ If this value were desired, it could be obtained by *minimizing* γ subject to the the restrictions (2), (3') and (4) and the imposed constraints $x_1 = 1, x_3 = 4$.

(A.1)

A	B	I_m	b
c'_A	c'_B	$0'$	$c_b = 0$

functional row

where I_m is the $m \times m$ unit matrix, A is a possible basis, and, allowing for the appropriate reordering of the columns:

A	B
c'_A	c'_B

 $=$

P	Q
c'	\bar{c}'

If the entire matrix (A.1) is premultiplied by

$$\begin{bmatrix} A & 0 \\ c'_A & -1 \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ c'_A A^{-1} & -1 \end{bmatrix}$$

the result is:

inverse
basis

(A.2)

I_m	$A^{-1}B$	A^{-1}	$A^{-1}b$
$0'$	$c'_A A^{-1}B - c'_B$	$c'_A A^{-1}$	$c'_A A^{-1}b$

functional row

When γ is added as a constraint, the basis corresponding to A for the enlarged problem is:

$$\left[\begin{array}{c|c} A & 0 \\ \hline c'_A & 1 \end{array} \right]$$

with inverse:

(A.3)

$$\left[\begin{array}{c|c} A^{-1} & 0 \\ \hline -c'_A A^{-1} & 1 \end{array} \right]$$

A unit column, with unit entry in the functional row is inserted in (A.1) between the inverse basis and the b column. The functional row now will have the form of the new constraint if the entry c_b is put equal to the current value of γ , i.e., if A is the current basis for the initial problem, $c_b = c'_A A^{-1}b$.

The convex set of feasible solutions to the initial problem is governed by the structure of the first m rows of (A.1). In the enlarged problem, the augmented matrix (A.1) plays the same role, and the result of multiplying this by (A.3) is:

(A.4)

inverse basis				
I_m	$A^{-1}B$	A^{-1}	0	$A^{-1}b$
0	$c'_B - c'_A A^{-1}B$	$-c'_A A^{-1}$	1	0

Provided that A yielded a feasible solution to the initial problem, (A.4) represents a feasible solution to the enlarged problem for which as yet no functional has been specified. In particular, if A is an optimal basis for the initial problem, the entries in the bottom row of (A.4) which, with the exception of the elements in the last column of the inverse basis and in the b column, are obtained by multiplying the functional row of (A.2) by -1 , are all nonpositive.

Throughout the calculation it is necessary to compute only the inverse basis and those rows of the rest of the tableau associated with the variable going out of the basis and with the x variable for which the functions $\min x$ and $\max x$ are currently being computed. In the larger problems summarised in Appendix 2, this represents a great economy over working with the whole tableau.

In the numerical example, the inverse basis and b column of (A.4) at the end of the solution of the initial problem are:

	I_1	I_2	I_3	I_4	b
x_1	0.0125	-0.0038	0.0088	0	1.4960
y_1	0.0401	0.0174	0.0176	0	6.1697
x_3	0.0220	0.0177	0.0213	0	5.0210
I_4	-6.8468	-2.9781	-4.2983	1	0

Columns I_1 and I_2 of the inverse basis correspond to the real variables y_3 and y_4 , respectively, and at a later stage may enter the basis at a positive level. The remaining two columns correspond to disposal variables on equality constraints and therefore may appear in the basis *only* at zero level.

The next steps in the calculations consist of tracing the functions $\min x$ and $\max x$ by means of parametric programming.

Min x_1 . In the enlarged problem, let the functional be x_1 and let the aim be to minimize this functional. The x_1 row becomes the functional row and the condition for a minimum will be satisfied if all the relevant elements in this row are negative. It can be seen that this requirement does not apply to columns I_3 and I_4 neither of which is a possible candidate for coming into the basis. The entire x_1 row¹² is calculated by premultiplying the matrix

$$\begin{bmatrix} P & Q \\ c' & \bar{c}' \end{bmatrix}$$

¹² Note that elements outside the inverse basis are not required to a very high degree of accuracy—only sufficient to enable a clear choice of a new basic variable to be made by the ratio rule given below. The only exception is the column corresponding to the new vector to be brought in when a change of basis has to be made, all the elements of which must be calculated to the same degree of accuracy as the inverse.

by the x_1 row vector of the inverse

$$[0.0125 \quad -0.0038 \quad 0.0088 \quad 0]$$

to give

$$\begin{array}{ccccccc} x_1 & x_2 & x_3 & y_1 & y_2 & y_3 (=I_1) & y_4 (=I_2) \\ [1 & 0.524 & 0 & 0 & 0.334 & 0.0125 & -0.0038] \end{array}.$$

Not all the entries are negative, so a basis change is required. The variable which is to go out of the basis is I_4 , the elements of which are¹³

$$\begin{array}{ccccccc} x_1 & x_2 & x_3 & y_1 & y_2 & y_3 & y_4 \\ [0 & -341.4 & 0 & 0 & -76.5 & -6.8468 & -2.9781] \end{array}.$$

The variable chosen to come in is y_2 since this ensures that all elements in the new x_1 row except the I_4 column are negative.¹⁴ When this change has been made, the I_4 column and the b column are:

	I_4	b
x_1	0.0044	1.4960
y_1	0.0060	6.1697
x_3	0.0054	5.0210
y_2	-0.0131	0

If now γ is decreased by d units, the elements b_i of the b column will alter according to the rule,

$$b_i^* = b_i - dI_{4i} \quad (i = 1, 2, \dots, 4),$$

and, in particular

$$x_1^* = 1.4960 - d(0.0044).$$

Therefore the value $\gamma_1(1)$ of γ for which the line $x_1 = 1$ intersects the function $\min x_1$, cannot exceed

$$1165 - d_{\min} = 1165 - \frac{0.4960}{0.0044} = 1051.8952.$$

If any of the other variables should become negative for some value of d smaller than d_{\min} , a change of basis will be required. The variable which is brought in as the result of a change of this type is chosen so as to maintain the optimal character of the functional row by a rule of the same nature as that described in the footnote. From the convexity of the set in Figure 5, such a change reduces the gradient of $\min x_1$ as a function of γ and $\gamma_1(1)$ would then be less than 1051.8952.

The remaining elements of the inverse basis and the b column need be calculated only if it becomes necessary actually to carry out the change of basis concerned.

If the entry in the I_4 column and x_1 row had been negative, this would have demonstrated the impossibility of reducing x_1 as a function of falling γ and would mean that the constraint $x_1 = 1$ was not feasible.

¹³ From the solution to the initial problem.

¹⁴ The variable chosen is the one for which $-x_{1j}/I_{4j}$ reaches its maximum value. Similarly, if $\max x_1$ were under consideration, the choice would be governed by the maximum value of x_{1j}/I_{4j} .

A similar method is used to determine the value of γ for the max x_1 function. The value of γ corresponding to $x_1 = 0$ has also to be determined. An upper bound to it can be obtained from the min x_1 calculation:

$$1165.5060 - \frac{1.4960}{0.0044} = 822.8457.$$

Since neither γ_1 nor x_3 has decreased to zero for this value of γ , this is exactly the value of γ for which $x_1 = 0$.

γ^1 is selected as the starting point for Step 4 and the complete inverse basis and b column for $\gamma = \gamma^1 = 1051.8952$ are worked out. An extra constraint, $x_1 - F = 1$, where F is a disposal variable which must always be zero, is now added to the programme. Theoretically this adds a row (F) and a column (I_5) to the inverse basis. The variable F (which is at present in the basis) plays the same role at this stage of the calculations as I_4 did in the previous stage. It can easily be shown that apart from the column I_5 the elements in the F row of the inverse basis are exactly the same as the elements of the x_1 row and therefore as soon as any basis change is made, the elements of the x_1 row in columns I_1 to I_4 vanish. It is now impossible for any basis change to affect the x_1 row and hence both it and the column I_5 may be dropped from the basis. The x_1 column in the main part of the tableau may also be deleted, since x_1 is now constrained to be equal to one.

In the succeeding steps of the calculation, the same method is employed to trace out the functions min x and max x , care being taken always to work down from the greatest value of γ which is still compatible with the constraints so far imposed. In the small numerical example, the solution was found without having to examine more than one branch of the tree, but in general this will not occur nor, of course, is there any guarantee that a solution exists.

Towards the end of a calculation following the second method, it may be helpful to use the first method to investigate some of the branches. The value of the first method is most apparent when the integral constraints have reduced the entries in the b column so much that a solution can almost be found by inspection.

Although the dual method for adding constraints used by Markowitz and Manne [6] can be adapted to the requirements of this algorithm, it appears to involve more work than the parametric method described above.

APPENDIX 2

The methods described in this paper have been tried out on the data published by Markowitz and Manne [6] as a hypothetical production problem. The problem they solved is:

$$\begin{aligned} &\text{Maximize } c'x = \gamma, \\ &\text{subject to } Ax \leq b, \\ &\quad x_j = 0 \text{ or } 1, \end{aligned}$$

where A is a 6×21 matrix of technical coefficients, all of which are positive integers ranging from 3 to 99.

An alternative problem restricts the variables only to being nonnegative integers:

$$x_j = \text{any nonnegative integer.}$$

"ANY INTEGER" PROBLEM

We have solved both problems, but have chosen to present only the second in any detail since it is the more general problem. The main description of the calculations is the "tree" figure (Figure 8, p. 518), the text being in the nature of an annotation to the tree.

The solution of the primary problem is:

$$\begin{aligned} &\text{Maximize } c'x = \gamma, \\ &\text{subject to } Ax \leq b, \\ &\quad x \geq 0, \\ \gamma^0 &= 643.99,^{15} \\ x_8 &= 1.77, \\ x_{11} &= 2.54, \\ x_{12} &= 1.04, \\ x_{13} &= 2.06, \\ x_{15} &= 0.97. \end{aligned}$$

It will be seen that x_{11} is the value furthest from an integer. In fact the min and max functions were computed for all 5 variables and x_{11} provided the least upper bound. Thus in this case the a priori selection of x_{11} as the variable to be considered would have given the same result as considering all variables. This is not always the case, however, and on the figure some γ 's will be found marked with an asterisk showing that in the following step the x variable is not the one which would have been chosen following the "furthest from an integer" rule. During the later calculations this rule was followed, and the min and max functions were not obtained for every x variable. The calculations followed the procedure described as Method 2 in Appendix 1 except that where the b values had been reduced so much that only a few integer solutions were possible (i.e., when the sum of the constraints reached 5, 6 or 7) Method 1 was tried. Since, in general, Method 1 discovers an interior point of the convex set enclosed by the min and max functions, successively higher integer values of the x variable being examined must be explored until an integer is reached for which the constraints are no longer feasible. These nonfeasible branches are omitted from Figure 8. The other branches for which Method 1 was used are marked M.1. and for these the highest integer solution compatible with the constraints has been determined and is shown. The γ values on the other branches are upper bounds on the value of the functional, obtained by Method 2.¹⁶

Two optimum solutions were discovered at $\gamma = 594$. These are:

1st solution		2nd solution	
1	= x_5		
1	= x_6		
		x_8	= 2
2	= x_{11}	= 1	
1	= x_{12}	= 2	
2	= x_{13}	= 1	
1	= x_{15}	= 1	
		x_{20}	= 1

¹⁵ Results have been rounded to two decimal places.

¹⁶ There may be other nonfeasible constraints for which an upper bound is shown. This merely means that the Method 2 calculation has not been taken far enough to show up the infeasibility.

It will be seen from the tree that the solution of the problem involved 37 steps (apart from the primary solution). This involved exploring 74×7 inverse bases, in

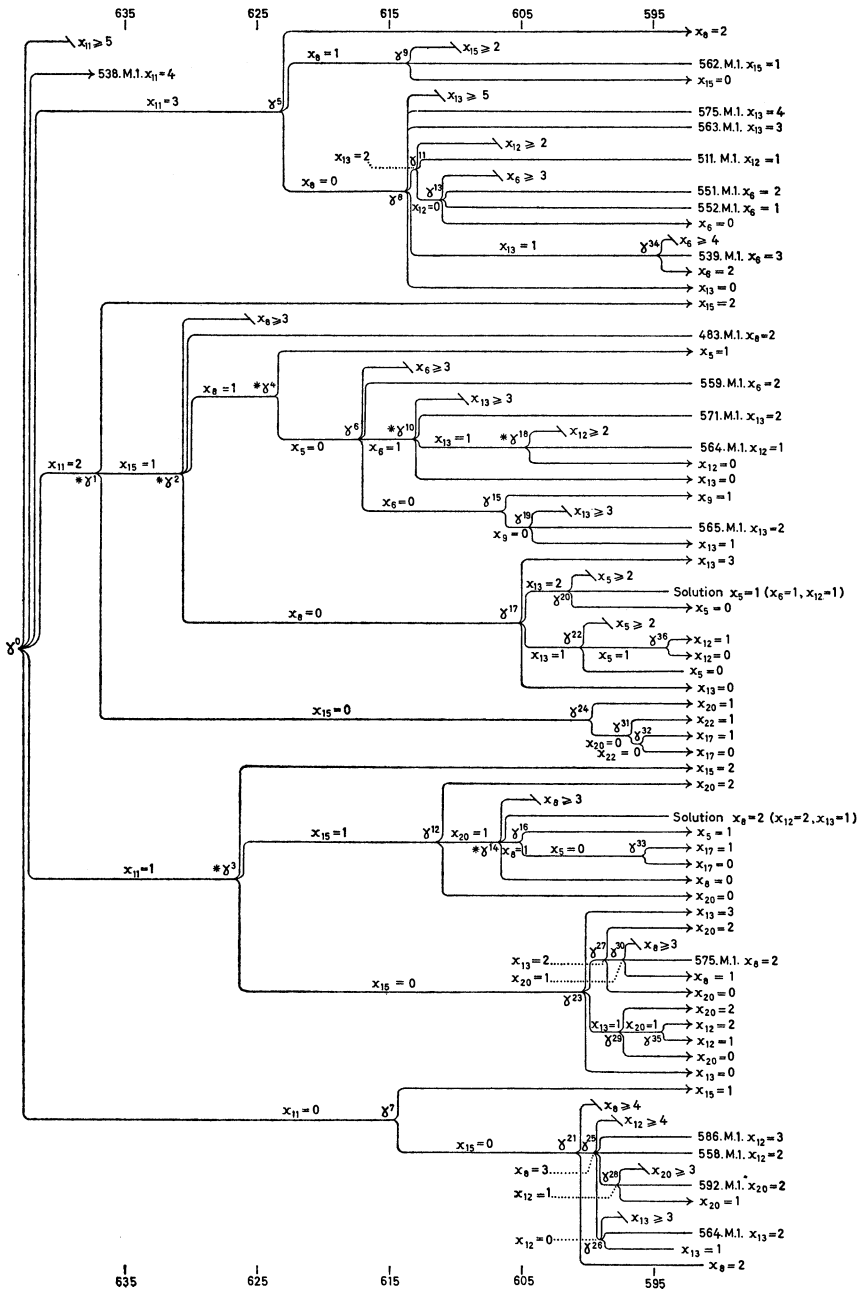


FIGURE 8

25 of which only the γ column was needed, and using Method 1 39 times, including 19 times in which the constraints were not feasible.

"0 OR 1" PROBLEM

The primary problem in this case was:

$$\begin{aligned} \text{Max } c'x &= \gamma, \\ \text{subject to } Ax &\leq b, \\ 0 &\leq x \leq 1. \end{aligned}$$

Dantzig's method for bounded variables [2] was used so that here again the inverse bases were of order 7×7 . The primary solution is

$$\begin{aligned} \gamma^0 &= 594.40, \\ x_5 &= 0.69, \\ x_6 &= 1, \\ x_8 &= 1, \\ x_9 &= 0.29, \\ x_{11} &= 1, \\ x_{12} &= 1, \\ x_{13} &= 1, \\ x_{15} &= 0.94, \\ x_{20} &= 0.96. \end{aligned}$$

It will be noted that even in the primary solution there are 5 variables already at the level one, and it remains true throughout the calculation that at each γ value there are not only variables which are constrained to be equal to zero or one, but also others which are at the value one because of the constraints $x_j \leq 1$. This has the unfortunate result that Method 1 cannot be used even when there are seven variables at the unit level, since they are not constrained to be equal to one, and therefore cannot be subtracted from the b column. Indeed at one point when approaching an integer solution at $\gamma = 538$ (the optimum being $\gamma = 540$) every unused x variable had to be successively constrained to be equal to zero before the upper bound of γ on that branch could be pushed below 540. This is not quite so disastrous as it sounds since in this situation one simply explored a series of bases with only two variables other than slack variables.

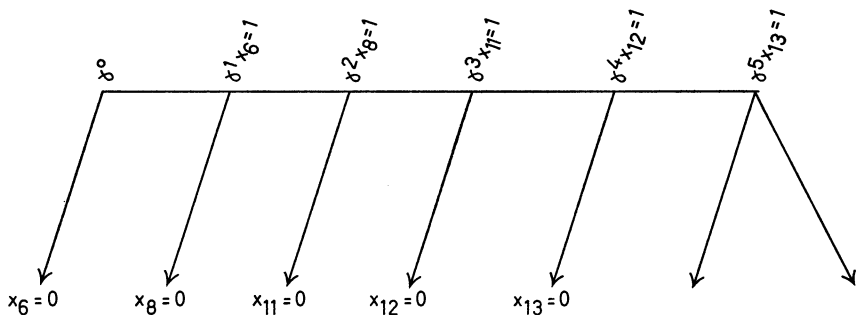


FIGURE 9

In summary there were 53 steps in this calculation. The total number of different bases was 165 in which 2 had 6 slack variables, 42 had 5, 63 had 4, 41 had 3, 15 had 2, and 2 had only 1. In 42 of the bases only the γ column was computed.

It appears to us almost certain that a different criterion for selecting the x variable at each step would have considerably reduced this work. In particular it appears that one should select not merely the greatest noninteger variable but the greatest unconstrained variable. In the problem starting with bounded variables, in other words, one would start by selecting (arbitrarily) x_6 , x_8 , x_{11} , x_{12} , or x_{13} and investigate the effect of reducing it to zero. The other side of the fork would be constraining it to be equal to unity, and hence would not reduce γ at all. The start of the tree, therefore, would appear as:

The testing of this hypothesis, however, will be postponed until it is programmed for an electronic computer.

Acknowledgement. We wish to thank Mr. Neil Swan for his valuable assistance in solving the numerical examples.

London School of Economics and Political Science

REFERENCES

- [1] BEALE, E. M. L.: "A Method of Solving Linear Programming Problems When Some But Not All of the Variables Must Take Integral Values," Statistical Techniques Research Group Technical Report No. 19, Princeton, N. J., July, 1958.
- [2] DANTZIG, G. B.: "Recent Advances in Linear Programming," *Management Science*, 2, (1956), pp. 131–44.
- [3] DANTZIG, G. B.: "Discrete-Variable Extremum Problems," *Operations Research*, 5, (1957), pp. 266–77.
- [4] GASS, S. I. AND T. L. SAATY: "Parametric Objective Function, II, Generalisation," *Journal of the Operations Research Society of America*, 3, (1955), pp. 395–401.
- [5] GOMORY, R. E.: "Outline of an Algorithm for Integer Solutions to Linear Programs," *Bulletin of the American Mathematical Society*, 64, (1958), pp. 275–8.
- [6] MARKOWITZ, H. M. AND A. S. MANNE: "On the Solution of Discrete Programming Problems," *Econometrica*, 25, (1957), pp. 84–110.
- [7] ORCHARD-HAYS, W.: "Revisions and Extensions to the Simplex Method," RAND Corporation, P-562, 2 September, 1954.
- [8] SAATY, T. L. AND S. I. GASS: "Parametric Objective Function, I," *Journal of the Operations Research Society of America*, 2, (1954), pp. 316–9.