REVUE FRANÇAISE D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE. SÉRIE ROUGE

G. MITRA
D. B. C. RICHARDS
K. WOLFENDEN

An improved algorithm for the solution of integer programs by the solution of associated diophantine equations

Revue française d'informatique et de recherche opérationnelle. Série rouge, tome 4, nº R1 (1970), p. 47-60

<http://www.numdam.org/item?id=M2AN_1970__4_1_47_0>

R.I.R.O.
(4e année, N° R-1, 1970, p. 47-60)

AN IMPROVED ALGORITHM FOR THE SOLUTION OF INTEGER PROGRAMS BY THE SOLUTION OF ASSOCIATED DIOPHANTINE EQUATIONS

by G. Mitra, D. B. C. Richards and K. Wolfenden (1)

Résumé. — *L'algorithme du type « Cutting plane » qui est élaboré ici semble avoir quelques avantages sur les algorithmes actuels du même type. Cet algorithme se sert d'un autre, le « Positive Diophantine », qui donne la solution d'une équation diophantine à variables non négatives. De ceux-là se développe une nouvelle méthode, méthode directe, qui donne les solutions des programmes avec des valeurs entières. Trois exemples détaillés illustrent la technique. L'algorithme « Positive Diophantine » se trouve à l'appendice.*

INTRODUCTION

In the course of general studies on techniques of integer programming an algorithm of the cutting plane type has been developed which appears to offer certain advantages over existing algorithms of the same type ([2], [3]). This algorithm makes use of another, « Positive Diophantine », for the solution of diophantine equations in non-negative variables. Together these have led to the development of a further method, a direct method similar to the technique outlined in [5].

1. THEORY

1.1. The basic problem

Consider the problem of maximizing

$$x_0 = a_{00} + \sum_{j=1}^{n} a_{0j}(-x_j) \tag{1.1.1}$$

(1) University of London Institute of Computer Science.

subject to

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant a_{i0} \qquad i = 1, 2, ..., m \tag{1.1.2}$$

and

$$x_j \geqslant 0, x_j \equiv 0 \bmod (1) \qquad j = 0, 1, 2, ..., n,$$

where for convenience it is assumed that all $a_{ij} \equiv 0 \bmod (1)$.

Introducing slack variables

$$x_i \equiv 0 \bmod (1), x_i \geqslant 0, \qquad i = n+1, n+2, ..., n+m,$$

and adding a set of trivial equations $x_j = -(-x_j), j = 1, 2, ..., n$, leads to the Tucker-Beale system

$$\left.\begin{aligned} x_0 &= a_{00} + \sum_{j=1}^{n} a_{0j}(-x_j) \\ x_s &= 0 + \sum_{j=1}^{n} -\delta_{sj}(-x_j) \qquad s = 1, 2, ..., n \\ x_i &= a_{i0} + \sum_{j=1}^{n} a_{ij}(-x_j) \qquad i = n+1, n+2, ..., n+m, \end{aligned}\right\} \tag{1.1.3}$$

where δ_{sj} is the Kronecker delta.

1.2. The continuous solution

Let $C(\bar{A})$ represent the continuous optimum solution to this problem as obtained by a simplex algorithm. This solution is contained within the form

$$\left.\begin{aligned} Dx_0 &= \bar{a}_{00} + \sum_{p \notin B} \bar{a}_{0p}(-x_p) \\ Dx_i &= \bar{a}_{i0} + \sum_{p \notin B} \bar{a}_{ip}(-x_p) \qquad i = 1, 2, ..., n+m, \end{aligned}\right\} \tag{1.2.1}$$

where B denotes the set of indices of the vectors forming the current basis, namely, the basis of the optimal program $C(\bar{A})$. The coefficients D, the modulus of the determinant of the current basis, and $\bar{a}_{ij}$, the elements of the matrix $\bar{A}$, are all integral. Setting the non-basic variables x_p to zero, the continuous optimum solution is $x_i = \bar{a}_{i0}/D$.

Let $I(\bar{A})$ denote the optimal integer program of the stated problem. If $\bar{a}_{i0} \not\equiv 0 \bmod (D)$ for any $i = 0, 1, 2, ..., m+n$, then $I(\bar{A}) \not\equiv C(\bar{A})$.

1.3. Cutting plane steps

Let e be the index of any row of the system (1.2.1) such that $\bar{a}_{ep} \not\equiv 0 \bmod (D)$ for at least one $p \notin B$. For such a row define the fractional elements

$$\left.\begin{aligned} f_p &= \frac{\bar{a}_{ep}}{D} - \left[\frac{\bar{a}_{ep}}{D}\right], \quad p \notin B, \quad \text{so that} \quad 0 \leqslant f_p < 1, \\ \text{and} \qquad f_0 &= \frac{\bar{a}_{e0}}{D} - \left[\frac{\bar{a}_{e0}}{D}\right], \quad 0 \leqslant f_0 < 1. \end{aligned}\right\} \tag{1.3.1}$$

As in [3] this leads to the introduction of the reduced inequality

$$f_0 \leqslant \sum_{p \notin B} f_p x_p \tag{1.3.2}$$

which must be satisfied by any feasible solution to the problem. However, if the fractional parts are expressed as ratios of integers with common denominator D, then under certain conditions a stronger inequality can be constructed. From (1.3.1) we can obtain non-negative integers d_0, d_p such that

$$f_0 = \frac{d_0}{D}, \qquad f_p = \frac{d_p}{D}, \qquad 0 \leqslant d_0, d_p < D, \tag{1.3.3}$$

and substituting in (1.3.2)

$$d_0 \leqslant \sum_{p \notin B} d_p x_p. \tag{1.3.4}$$

Of course, this is just one of a series of parallel « cuts » given by

$$d_0 + rD \leqslant \sum_{p \notin B} d_p x_p \qquad r = 0, 1, 2, \ldots, \tag{1.3.5}$$

and such that the larger the value of r the deeper the cut into the convex region (1.1.2). It would appear desirable, therefore, to be able to determine the largest possible value of r such that the corresponding cut does not exclude any feasible lattice point. However, the best that can be done is to find the minimum value of r for which the diophantine equation

$$d_0 + rD = \sum_{p \notin B} d_p x_p$$

has a non-negative integer solution in the variables x_p. Note that in the case of a primary cut ($r = 0$) the diophantine equation always admits of solution in integer variables unrestricted in sign.

Formally, the problem now is to find minimum r and $x_p \equiv 0 \bmod (1)$, $x_p \geqslant 0$ such that

$$\sum_{\notin B} d_p x_p = d_0 + rD. \tag{1.3.6}$$

A dynamic programming solution has been proposed in [2] which, though conceptually elegant, is hardly computationally efficient. An alternative approach is provided by the algorithm « Positive Diophantine » (Appendix). The aim is to locate lattice points on the finite parallel planes (1.3.6) within the bounds $0 \leqslant x_p \leqslant \left[\dfrac{d_0 + rD}{d_p}\right]$. In most cases where cuts with $r > 0$ (secondary cuts) have been found to exist, the convergence to $I(\bar{A})$ has been more rapid than with primary cuts. Computationally worthwhile improvement has been observed in a number of test problems (Table 1), only two taking more iterations than with Gomory's method.

1.4. Direct determination of $I(\bar{A})$ from $C(\bar{A})$

It is possible to proceed from $C(\bar{A})$ to $I(\bar{A})$ in the manner of [3] and at each cut generation stage to apply the process outlined in the previous section. However, the additional pivoting involved can be avoided by taking advantage of the search technique of Section 1.3.

Let
$$\bar{X} = \{ \bar{x}_i \mid \bar{x}_i = \bar{a}_{i0}/D, \; i \in B \; ; \; \bar{x}_i = 0, \; i \notin B \}$$

denote the solution $C(\bar{A})$ of the continuous problem. Then corresponding to an optimum integer solution X^I, there exists a non-negative integer vector of n components $x_p \equiv 0 \bmod (1)$, $p \notin B$, such that X^I can be expressed ([5]) as

$$X^I = \left\{ x_i^I \mid x_i^I = \bar{a}_{i0}/D + \sum_{p \notin B} (\bar{a}_{ip}/D)(-x_p), \; i \in B \, ; \; x_i^I = x_p, \; i = p \notin B \right\} \tag{1.4.1}$$

where of necessity $x_i^I \equiv 0 \bmod (1)$ and $x_i^I \geqslant 0$, $i = 1, 2, ..., n + m$. From (1.4.1) it follows that to obtain the integer program directly we need to determine the components of this all important n-vector of the variables x_p, $p \notin B$. To do this we can use the objective row of $\bar{A}$ without extracting the cut and investigate the parallel hyperplanes corresponding to different values of r until a feasible lattice point can be located on one of them. If, from the objective row of the matrix $\bar{A}$ we extract the congruence

$$\sum_{p \notin B} \bar{a}_{0p} x_p \equiv d_0 \bmod (D)$$

and rewrite it in the form

$$d_0 + rD = \sum_{p \notin B} \bar{a}_{0p} x_p \qquad x_p \equiv 0 \bmod (1), \qquad x_p \geqslant 0 \tag{1.4.2}$$

then, it is to be noted, (1.4.2) poses the same problem as (1.3.6), namely, solution in positive integers of a diophantine equation with positive coefficients ($\bar{a}_{0p}$ are all non-negative integers, since the optimal tableau is of necessity dual feasible (1)). For a fixed value of r, (1.4.2) defines a finite plane in the n-space of $x_p, p \notin B$, and all the lattice points on this plane are generated by the search « Positive Diophantine ». These in turn are substituted in the relation (1.4.1) and the results x_i^I, $i \in B$, are checked for non-negativity and integrality.

Assuming $C(\bar{A})$ is not dual degenerate, that is, the objective row of $\bar{A}$ does not contain any zero element, an algorithm for obtaining $I(\bar{A})$ from $C(\bar{A})$ is outlined below.

1. Extract (1.4.2) from (1.2.1) and set $r = 0$.

2. Apply the algorithm « Positive Diophantine » to explore whether there exist $x_p \equiv 0 \bmod (1)$ and $x_p \geqslant 0$, $p \notin B$, which satisfy the equality of (1.4.2).

3. Is the present hyperplane exhausted? If yes then $r := r + 1$; go to step 2.

4. $x_p \geqslant 0, p \notin B$, are the components of an integer vector satisfying (1.4.2). Does this vector substituted in (1.4.1) satisfy the integrality and non-negativity requirements of X^I? If yes then go to 6.

5. Go to 2.

6. Output the optimal integer solution and stop.

Note that the algorithm can be simply modified to produce alternative optima if they exist. Feasibility or otherwise of the current program for each successive value of r can likewise be established with little extra effort.

2. EXAMPLES

2.1. Generalized cut

Consider the problem (taken from [6]) :

$$\text{Minimize} \qquad x_2$$

$$\text{subject to} \qquad \begin{aligned} 33x_1 + 7x_2 &\geqslant 715 \\ -41x_1 + 14x_2 &\geqslant 653 \end{aligned} \tag{2.1.1}$$

$$x_j \equiv 0 \bmod (1) \quad \text{and} \quad x_j \geqslant 0 \quad j = 1, 2.$$

---

(1) For the time being we assume that $\bar{a}_{0p} > 0$ for all $p \notin B$. The problem of dual degeneracy is dealt with in Section 2.3.

Writing the tableau in the Tucker-Beale form and applying the dual simplex pivot rules we obtain the continuous optimum as follows :

Iteration 0

$D = 1$

		$-x_1$	$-x_2$
x_0	0	0	1
x_1	0	-1	0
x_2	0	0	-1
x_3	-715	-33^*	-7
x_4	-653	41	-14

Note : x_3, x_4 denote slacks; pivot element is starred.

Iteration 1

$D = 33$

		$-x_3$	$-x_2$
x_0	0	0	33
x_1	715	-1	7
x_2	0	0	-33
x_3	0	-33	0
x_4	$-50\,864$	41	-749^*

Iteration 2

$D = 749$

		$-x_3$	$-x_4$
x_0	$-50\,864$	41	33
x_1	5 439	-14	7
x_2	50 864	-41	-33
x_3	0	-749	0
x_4	0	0	-749

Continuous optimum.

Using the objective row to generate the cut, we have by relation (1.3.5)

$$68 + 749r \leqslant 41x_3 + 33x_4 \qquad (2.1.2)$$

Taking $r = 0$ for this and all the subsequent cutting planes (Gomory's method [3]), the optimum integer solution was obtained after 32 pivots steps.

However, applying « Positive Diophantine » to (2.1.2) as described in Section 1.3 we obtain $\min(r) = 2$, and append the corresponding generalized cut to Iteration 2.

Iteration 2

$D = 749$

		$-x_3$	$-x_4$
x_0	$-50\,864$	41	33
x_1	5 439	-14	7
x_2	50 864	-41	-33
x_3	0	-749	0
x_4	0	0	-749
s_1	$-(68 + 2 \times 749)$	-41	-33*

Note : A random choice is made to break the tie for the selection of pivot column.

Iteration 3

$D = 33$

		$-x_3$	$-s_1$
x_0	$-2\,310$	0	33
x_1	225	-1	7
x_2	2 310	0	-33
x_3	0	-33	0
x_4	1 566	41	-749
s_2	$-(27 + 5 \times 33)$	-32*	-7

Note : Second row is used for cut extraction. Applying « Positive Diophantine » we obtain $\min(r) = 5$ for which there exists a solution in positive integers for x_3 and s_1.

Iteration 4

$D = 32$

		$-s_2$	$-s_1$
x_0	$-2\,240$	0	32
x_1	224	-1	7
x_2	2 240	0	-32
x_3	192	-33	7
x_4	1 280	41	735

Note : The optimum integer solution obtained in 4 iterations is

$$\min x_2 = \frac{2\,240}{32} = 70$$

$$x_1 = \frac{224}{32} = 7$$

$$x_3 = \frac{192}{32} = 6$$

$$x_4 = \frac{1\,280}{32} = 40$$

2.2. Direct method

Consider the continuous optimum tableau of the previous problem (2.1.1).

$$D = 749$$

		$-x_3$	$-x_4$
x_0	$-50\,864$	41	33
x_1	$5\,439$	-14	7
x_2	$50\,864$	-41	-33
x_3	0	-749	0
x_4	0	0	-749

We extract the equation

$$68 + 749r = 41x_3 + 33x_4 \tag{2.2.1}$$

from this tableau as explained in Section 1.4 (note that this is the same as the cut extracted from the objective row in the first cutting plane step of the last section). Applying the algorithm of Section 1.4 to equation (2.2.1) we obtain min $(r) = 2$ and the corresponding solution

$$68 + 2(749) = 41(6) + 33(40),$$

$$\text{i.e.} \qquad x_3 = 6, \qquad x_4 = 40.$$

Substituting this in the continuous tableau as in (1.4.1) we have

$$\begin{aligned}
749(x_0) &= -50\,864 - 41(6) - 33(40) = -70(749) \equiv 0 \bmod (749) \\
749(x_1) &= 5\,439 + 14(6) - 7(40) = 7(749) \equiv 0 \bmod (749) \geqslant 0 \\
749(x_2) &= 50\,864 + 41(6) + 33(40) = 70(749) \equiv 0 \bmod (749) \geqslant 0 \\
749(x_3) &= 0 + 749(6) + 0 = 6(749) \equiv 0 \bmod (749) \geqslant 0 \\
749(x_4) &= 0 + 0 + 749(40) = 40(749) \equiv 0 \bmod (749) \geqslant 0
\end{aligned}$$

and the solution $I(\bar{A})$ has been obtained directly from $C(\bar{A})$.

Consider next the application of this direct method to another problem from [6] :

$$\begin{array}{lll} \text{Minimize} & x_3 & \\ \text{subject to} & 5x_1 + 8x_2 - 7x_3 \geqslant -89 & \\ & -6x_1 + 5x_2 + x_3 \geqslant 11 & x_j \equiv 0 \bmod (1), \quad x_j \geqslant 0 \\ & 3x_1 - 5x_2 + 2x_3 \geqslant 29 & j = 1, 2, 3. \end{array}$$

Iteration 0

$D = 1$

		$-x_1$	$-x_2$	$-x_3$
x_0	0	0	0	1
x_1	0	-1	0	0
x_2	0	0	-1	0
x_3	0	0	0	-1
x_4	89	-5	-8	7
x_5	-11	6	-5*	-1
x_6	-29	-3	5	-2

Iteration 1

$D = 5$

		$-x_1$	$-x_5$	$-x_3$
x_0	0	0	0	5
x_1	0	-5	0	0
x_2	11	-6	-1	1
x_3	0	0	0	-5
x_4	533	-73	-8	43
x_5	0	0	-5	0
x_6	-200	12	5	-15*

Iteration 2

$D = 15$

		$-x_1$	$-x_5$	$-x_6$
x_0	-200	15	5	5
x_1	0	-15	0	0
x_2	-7	-15	-2	1
x_3	200	-15	-5	-5
x_4	-121	-90*	19	43
x_5	0	0	-15	0
x_6	0	0	0	-15

Iteration 3

$D = 90$

		$-x_4$	$-x_5$	$-x_6$
x_0	$-1\,321$	15	49	73
x_1	121	-15	-19	-43
x_2	79	-15	-31	-37
x_3	1 321	-15	-49	-73
x_4	0	-90	0	0
x_5	0	0	-90	0
x_6	0	0	0	-90

Continuous optimum

Extracting the equation (1.4.2) from the objective row,

$$29 + 90r = 15x_4 + 49x_5 + 73x_6,$$

and applying the algorithm of Section 1.4, we obtain the minimum r for which there exists a solution, namely min $(r) = 3$. Then

$$29 + 3(90) = 15(2) + 49(4) + 73(1),$$

i.e.

$$x_4 = 2, \qquad x_5 = 4, \qquad x_6 = 1.$$

Substituting in the above tableau :

$$\begin{aligned}
90(x_0) &= -1\,321 - 15(2) - 49(4) - 73(1) = -18(90) \equiv 0 \bmod (90)\\
90(x_1) &= 121 + 15(2) + 19(4) + 43(1) = 3(90) \equiv 0 \bmod (90) \geqslant 0\\
90(x_2) &= 79 + 15(2) + 31(4) + 37(1) = 3(90) \equiv 0 \bmod (90) \geqslant 0\\
90(x_3) &= 1\,321 + 15(2) + 49(4) + 73(1) = 18(90) \equiv 0 \bmod (90) \geqslant 0\\
90(x_4) &= 0 + 90(2) + 0 + 0 = 2(90) \equiv 0 \bmod (90) \geqslant 0\\
90(x_5) &= 0 + 0 + 90(4) + 0 = 4(90) \equiv 0 \bmod (90) \geqslant 0\\
90(x_6) &= 0 + 0 + 0 + 90(1) = 1(90) \equiv 0 \bmod (90) \geqslant 0
\end{aligned}$$

which is the solution $I(\bar{A})$ to the integer problem.

Lastly, to bring out the relationship between the proposed method and dynamic programming, we consider a cargo-loading problem taken from [1].

Maximize

$$72x_1 + 60x_2 + 40x_3 + 27x_4 + 20x_5 + 50x_6 + 85x_7 + 96x_8$$

subject to

$$20x_1 + 18x_2 + 14x_3 + 12x_4 + 10x_5 + 16x_6 + 22x_7 + 24x_8 \leqslant 100,$$
$$x_1\ x_2, \ldots, x_8 \geqslant 0 \quad \text{and} \quad x_1, x_2, \ldots, x_8 \equiv 0 \bmod (1).$$

Using the reduced tableau, that is, omitting the unit rows, we have

Iteration 0

$D = 1$

		$-x_1$	$-x_2$	$-x_3$	$-x_4$	$-x_5$	$-x_6$	$-x_7$	$-x_8$
x_0	0	-72	-60	-40	-27	-20	-50	-85	-96
x_9	100	20	18	14	12	10	16	22	24*

Iteration 1

$D = 24$

		$-x_1$	$-x_2$	$-x_3$	$-x_4$	$-x_5$	$-x_6$	$-x_7$	$-x_9$
x_0	9 600	192	288	384	504	480	336	72	96
x_8	100	20	18	14	12	10	16	22	1

We now investigate by " Positive Diophantine " the problem of minimizing r subject to non-negative integral solution of

$$0 + r(24) = 192x_1 + 288x_2 + 384x_3 + 504x_4 + 480x_5 + 336x_6 + 72x_7 + 96x_9.$$

It is found that no such solution exists for $r = 0, 1, 2$ whereas for $r = 3$ the solution (0, 0, 0, 0, 0, 0, 1, 0) is unique. Substituting in the constraint equation we have

$$x_8 = 100 - 22(1) = 78 \not\equiv 0 \bmod (24).$$

Next, $r = 4$ has the solution (0, 0, 0, 0, 0, 0, 0, 1) but again the integrality requirement of the constraint equation is not satisfied,

$$x_8 = 100 - 1(1) = 99 \not\equiv 0 \bmod (24).$$

The cyclic process of incrementing r, determining lattice points and testing the constraint is continued until $r = 16$. The solutions of the diophantine equation are now (2, 0, 0, 0, 0, 0, 0, 0), (1, 0, 0, 0, 0, 0, 0, 2), (0, 0, 0, 0, 0, 0, 0, 4). The first two solutions do not satisfy the integrality condition for x_8 but the last one does,

$$x_8 = 100 - 1(4) = 96 \equiv 0 \bmod (24).$$

Hence

$$x_8 = 4, \qquad x_1 = x_2 = \dots = x_7 = 0$$

is the optimum solution to the problem with $x_0 = 384$.

2.3. Results and Conclusions

In all some 30 problems were tackled by both the optimum cut method and Gomory's Method of Integer Forms on the ICT Atlas at the Institute of Computer Science. These problems, taken from a number of sources, have all been rated as difficult integer programming problems in some sense or other. Several variants of the two main Fortran programs were written to accommodate different cut generator selection rules. Complete results for a matched pair of programs are given below; overall execution times for the optimum cut method and Gomory's Method were 65.0 secs and 85.7 secs

TABLE 1

PROBLEM No.	DIMENSIONS		NO. OF ITERATIONS TO INTEGER SOLUTION (including the continuous stage)	
	Constraints	Variables	Gomory's Method	Optimum Cut Method
1	4	5	25	8
2	4	5	18	18
3	4	5	43	19
4	4	5	13	7
5	6	5	1 431	397
6	6	5	80	80
7	4	5	> 2 004	557
8	4	5	89	81
9	6	6	558	469
10	4	5	8	8
11	7	7	9	9
12	7	7	25	25
13	3	4	17	19
14	15	15	41	29
15	2	2	37	35
16	3	3	30	7
17	2	2	13	4
18	3	3	15	6
19	3	3	55	33
20	2	2	200	200
21	2	2	305	89
22	2	2	298	86
23	8	8	46	48
24	2	2	10	6
25	3	2	5	5
26	5	2	8	8
27	3	3	4	4
28	2	5	4	4
29	5	2	12	12
30	3	3	9	5

respectively, the latter including a problem which remained unsolved after the imposed limit of 2 000 iterations.

Although our experience with both the optimum cut method and the related direct method is limited (program development with the direct method is as yet incomplete) the following points should be noted.

1) The D-number can become too large for single length arithmetic even before the continuous optimum $C(\bar{A})$ is reached, thus inhibiting the application of either method. To combat this an adaptative cut generation technique [4] which attempts to restrain the growth of D might be applied.

2) In the direct method, if the continuous optimum solution is dual degenerate then there will exist an infinity of solutions for the diophantine equation extracted from the objective row, namely

$$d_0 + rD = \bar{a}_{01}x_1 + \bar{a}_{02}x_2 + \ldots + \bar{a}_{0n}x_n \tag{2.3.1}$$

For if $\bar{a}_{0i} = 0$ for at least one i, $1 \leqslant i \leqslant n$, x_i can assume any non-negative integral value. This difficulty might be resolved by holding the already determined components of a solution to (2.3.1) at their current values and varying the parametric values of the indeterminate components (corresponding to $\bar{a}_{0i} = 0$) in a lexicographically ordered search on the successive equations.

APPENDIX. THE ALGORITHM "POSITIVE DIOPHANTINE"

Consider the congruence

$$\sum_{i=1}^{n} \alpha_i x_i \equiv \alpha_0 \bmod (D) \tag{A.1}$$

where $\alpha_i \equiv 0 \bmod (1)$ and $\alpha_i > 0$, $x_i \equiv 0 \bmod (1)$ and $x_i \geqslant 0$, $i = 1, 2, \ldots, n$. For convenience let us assume that the coefficients are ordered, so that

$$\alpha_1 \leqslant \alpha_2 \leqslant \alpha_3 \ldots \leqslant \alpha_n.$$

Let the diophantine equation

$$\sum_{i=1}^{n} \alpha_i x_i = \alpha_0 + rD = \Phi_n \tag{A.2}$$

define the problem P_n. Then set $x_n = \left[\dfrac{\Phi_n}{\alpha_n}\right]$, so that the remaining x_i must satisfy

$$\sum_{i=1}^{n-1} \alpha_i x_i = \Phi_n - \alpha_n\left[\frac{\Phi_n}{\alpha_n}\right] = \Phi_{n-1},$$

defining the problem P_{n-1}. Repetition of the procedure leads to

$$\alpha_1 x_1 = \Phi_2 - \alpha_2 \left[\frac{\Phi_2}{\alpha_2} \right] = \Phi_1,$$

defining the problem P_1, and we define

$$\Phi_1 - \alpha_1 \left[\frac{\Phi_1}{\alpha_1} \right] = \Phi_0.$$

If at any stage Φ_k is zero, the congruence (A.1) is satisfied by the x_i already set for $i \geqslant k + 1$, and $x_i = 0$ for $i \leqslant k$. If the entire sequence P_i, $i = n$, $n - 1, \ldots, 1$, is produced and $\Phi_0 \neq 0$, then a new subsequence must be defined, starting at the last i for which $x_i > 0$. Suppose this was $i = k$, then reset $x_k = x_k - 1$ and $\Phi_{k-1} = \Phi_{k-1} + \alpha_k$ and proceed with the new sub-problems P_i, $i = k - 1$, $k - 2, \ldots, 1$. Whenever $\Phi_i = 0$, the corresponding n-vector is a lattice point on the hyperplane defined by (A.2). The recursive process can be continued until the search over the finite hyperplane is complete, that is, the components $x_n, x_{n-1}, \ldots, x_2$ are reduced to zero.

REFERENCES

[1] BELLMAN, R. E. and DREYFUS, S., *Applied Dynamic Programming* (Princeton University Press, 1962), pp. 27-31.

[2] FINKELSHTEYN, Yu. Yu., *Additional Restrictions for Problems of Integer Linear Programming*, Engineering Cybernetics (Translated from Russian), May-June 1965, pp. 38-39.

[3] GOMORY, R. E., *An Algorithm for Integer Solutions to Linear Programs*, *Recent Advances in Mathematical Programming*, ed. Graves, R. L. and Wolfe, P. (McGraw Hill, 1963), pp. 269-302.

[4] GOMORY, R. E., *An All-Integer Integer Programming Algorithm*, *Industrial Scheduling*, ed. Muth, J. F. and Thompson, G. L. (Prentice Hall, 1963), pp. 193-206.

[5] GOMORY, R. E., *On the Relation between Integer and Non-Integer Solutions to Linear Programs*, Proceedings of National Academy of Sciences (U.S.), vol. 53 (1965), pp. 260-265.

[6] THOMPSON, G. L., *The Stopped Simplex Method — 1 Basic Theory for Mixed Integer Programming*, *Integer Programming*, Revue Française de Recherche Opérationnelle, vol. 8 (1964), pp. 159-182.