NEW EFFICIENT ALGORITHM TO SOLVE PURE INTEGER PROGRAMMING PROBLEM

G.Karthikeyan,
Assistant Professor in Computer Science,
Government Arts College, Paramakudi,
Tamilnadu,India.

Prof S.Sakthivel,
Principal,
PSNA College of Engineering and Technology,
Dindugal, Tamilnadu,India.

Abstract: When formulating linear programming problem, variables should have been regarded as taking integer values. Problems in which this is the case are called integer program. In this paper a new algorithm to solve integer linear programming problems is given. This algorithm consists of two steps. In step 1 the intercepts of a promising variable based on the different constraints are found out. Using the intercept matrix obtained for all the promising variables, a maximum of m variables are selected and arranged where m is the number of constraints. Also the maximum value that each of the arranged variable can assume is found out. In step 2, the arranged variables are allowed to enter into the basis with an integer value which is less than the maximum value it can assume. Step 1 and 2 are repeated till no variable could enter with integer value. In the proposed integer linear programming algorithm the improved solution moves in the interior of the feasible region.

Keywords: Arrangement of variables, feasible solution, algorithm

I. INTRODUCTION

A linear programming problem in which some or all of the variables in the optimal solution are restricted to assume non-negative integer values is called an integer programming problem [I.P.P] or integer linear programming.

The general integer programming problem is given by

Extremize Z = CX

Subject to

$$AX \;\begin{matrix} < \\ = \\ > \end{matrix}\; P_0$$

$X \geq 0$ and all variables are integers.

Where

$$\underset{(mxn)}{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} ; \underset{(nx1)}{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} ; P_0 = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{bmatrix}$$

Let the columns corresponding to the matrix A be denoted by $P_1, P_2, \ldots.. P_n$ where

$$\underset{(mx1)}{P_1} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{m1} \end{bmatrix} \text{ and } P_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ a_{3n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

II. EXISTING METHODS

Some of the existing algorithms to solve linear integer programming problems which are amenable to computer are

1. Branch and Bound algorithms
2. Zero-one implicit enumeration algorithms
3. Cutting plane algorithms

The branch and bound algorithm is linear programming based tree search. This is a way of systematically enumerating feasible solution such that optimal integer solution is found.

There is an alternative to branch and bound called *cutting planes* which can also be used to solve integer programs. The fundamental idea behind cutting planes is to add constraints to a linear program until the optimal basic feasible solution takes on integer values. A special type of constraint called a *cut* is added to the problem. A cut relative to a current fractional solution satisfies the following criteria:

1. Every feasible integer solution is feasible for the cut, and
2. The current fractional solution is not feasible for the cut.

There are two ways to generate cuts. The first, called Gomory cuts, generates cuts from any linear programming tableau. This has the advantage of ``solving" any problem but has the disadvantage that the method can be very slow. The second approach is to use the structure of the problem to generate very good cuts. The approach needs a problem-by-problem analysis, but can provide very efficient solution techniques than Gomory cuts.

This kind of univariate search approach increases the number of iterations and computational effort. This leads to device methods to reduce the computational effort and time in converging the optimum solutions.

III.PROPOSED ALGORITHM

In this paper a new algorithm to solve integer linear programming problems is given. This algorithm consists of two Phases.

The step by step procedures of the proposed algorithms are as given below.

Step 1: Order the promising variables as given in Phase I.

Step 2: In this any variable in the set J, if yes step 2 else step 3 allows the arranged variables to enter in the basis. Go to step 1.

Step 3: Report the result and stop.

a) Phase I - Ordering of Promising Variables

Step 1: The matrix of intercepts of the decision variables along the respective axes called "θ" matrix with respect to the chosen basis is to be constructed. A typical intercept for the j^{th} variable, x_j due to the i^{th} the resource, b_i is

$$\frac{b_i}{a_{ij}}, a_{ij} > 0$$

The expanded form of θ matrix is :

$$\begin{array}{c} \\ x_1 \\ x_2 \\ \\ x_j \\ \\ x_n \end{array} \begin{array}{c} \begin{array}{cccccc} s_1 & s_2 & & s_i & & s_m \end{array} \\ \begin{bmatrix} b_1/a_{11} & b_2/a_{21} & \cdots & b_i/a_{i1} & \cdots & b_m/a_{m1} \\ b_1/a_{12} & b_2/a_{22} & \cdots & b_i/a_{i2} & \cdots & b_m/a_{m2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ b_1/a_{1j} & b_2/a_{2j} & \cdots & b_i/a_{ij} & \cdots & b_m/a_{mj} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ b_1/a_{1n} & b_2/a_{2n} & \cdots & b_i/a_{in} & \cdots & b_m/a_{mn} \end{bmatrix} \end{array}$$

Each row of the θ matrix represents the m number of intercepts of the decision variables along their respective axes and the each column represents the intercepts formed by the decision variables of each of the m constraints.

Step 2: Scan each row of θ matrix and find the minimum intercept and its position. Multiply the minimum intercepts with the corresponding contribution coefficient (c_j) value.

Step 3: Find the variable whose c_jx_j value is the largest. Let it be x_R. Then x_R is selected as the promising variable. If more than one largest c_jx_j value occurs, consider the variable that has maximum contribution coefficient including the fractional value is selected as promising variable. Delete the x_R^{th} row as well as the other rows whose minimum occurs in the position at which the minimum for x_R occurs. If more than one minimum occurs consider the variable that has minimum coefficient value including the fractional part. R is stored as the k^{th} element in set J, increment k by 1.

Step 4: Repeat step 3 till all the rows or all the columns are deleted.

Step 5: The set of variables collected in step 3 are the ordered promising variables.

$$\text{Let } J = \left\{ \begin{array}{l} \text{Subscripts of the promising variables arranged} \\ \text{in the descending order of } (c_j x_j) \text{ value} \end{array} \right\}$$

b) Phase II - Arranged variables are allowed to enter into the basis

Step 1: The first variable x_j in the set is allowed to enter the basis with actual value if x_j =1 else $x_j = \alpha x_j$ value.

P_0 vector is modified using the formula

$$(P_{0new})_i = (P_{0old})_i - \alpha x_j \qquad i = 1,2,\ldots, m.$$

Replace (P_{0old}) by P_{0new}.

Let $R = 1$.

Where αx_j is an integer $0 \le \alpha \le 1$.

Step 2: If R< k then Let t be the subscript of the $(R+1)^{th}$ element is in set J and x_t be the corresponding variable else go to step 6.

Step 3: For the variable x_t

Compute

$$\theta_1 = \min\left\{ \frac{(P_{0old})_i}{(a_{ji})_i} ; a_{ji} > 0 \right\}$$

$$X_t = c_t \theta_1$$

$$X_j = c_j (1-\alpha) x_j$$

Step 4:

If $X_t \rangle X_j$

replace x_j by x_t

if $R \ge k$ go to step 2

Step 5:

Compute

$$(P_{0new})_i = (P_{0old})_i - \alpha x_j$$

where $0 \le \alpha \le 1$

increment R by 1

go to step 2.

Step 6: Repeat Phase I and II until $P_{0new} \leq 0$

3.3 Numerical example

A numerical examples have been solved to illustrate the above method.

$$\text{Max } Z = 2x_1 + 5x_2 + 7x_3 + 3x_4 + 8x_5 + 6x_6 + 9x_7 + 4x_8 + 3x_9 + 11x_{10}$$

Subject to

$$2x_1 + 3x_2 + 6x_3 + 4x_4 + x_5 + 7x_6 + 2x_7 + 5x_8 + 9x_9 + x_{10} \leq 100$$
$$x_1 + 9x_2 + 5x_3 + 7x_4 + 3x_5 + 4x_6 + 8x_7 + 6x_8 + 3x_9 + 2x_{10} \leq 120$$
$$4x_1 + 2x_2 + 6x_3 + 5x_4 + 6x_5 + x_6 + 3x_7 + 8x_8 + 4x_9 + 7x_{10} \leq 100$$
$$10x_1 + x_2 + 3x_3 + 5x_4 + 9x_5 + 5x_6 + 9x_7 + 6x_8 + 3x_9 + 8x_{10} \leq 130$$
$$8x_1 + 5x_2 + 7x_3 + 9x_4 + 2x_5 + 8x_6 + 7x_7 + 2x_8 + 4x_9 + x_{10} \leq 120$$

Construction of θ matrix

c_j	x						$c_j * x_j$	
2	x_1	50	120	25	13	15	26	
5	x_2	33	13	50	130	24	65	4
7	x_3	**16.67**	24	**16.67**	43	17	112	
3	x_4	25	17	20	26	13	39	5
8	x_5	100	40	16	14	60	112	
6	x_6	14	30	100	26	15	84	3
9	x_7	50	15	33	14	17	126	2
4	x_8	20	20	12	21	60	48	
3	x_9	11	40	25	43	30	33	
11	x_{10}	100	60	14	16	120	154	1

Arrangement of Promising variables $J = \{x_{10}, x_7, x_6, x_2, x_4\}$

$$\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{array} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \overset{x_{10}=14}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 7 \end{bmatrix}} \quad \overset{x_{10}=7}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix}} \quad \overset{x_6=10}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 5 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix}} \quad \overset{x_2=8}{\begin{bmatrix} 0 \\ 4 \\ 0 \\ 0 \\ 0 \\ 5 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix}} \quad \overset{x_2=4}{\begin{bmatrix} 0 \\ 6 \\ 0 \\ 0 \\ 0 \\ 5 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix}}$$

$$\qquad\quad \downarrow \qquad\quad \downarrow \qquad\quad \downarrow \qquad\quad \downarrow \qquad\quad \downarrow$$

$$P_0 = \begin{bmatrix} 100 \\ 120 \\ 100 \\ 130 \\ 120 \end{bmatrix} \quad \begin{bmatrix} 93 \\ 106 \\ 51 \\ 74 \\ 113 \end{bmatrix} \quad \begin{bmatrix} 90 \\ 100 \\ 30 \\ 50 \\ 110 \end{bmatrix} \quad \begin{bmatrix} 55 \\ 80 \\ 25 \\ 25 \\ 70 \end{bmatrix} \quad \begin{bmatrix} 43 \\ 44 \\ 17 \\ 21 \\ 50 \end{bmatrix} \quad \begin{bmatrix} 37 \\ 26 \\ 13 \\ 19 \\ 40 \end{bmatrix}$$

 e-ISSN: 2395-5325

Construction of θ matrix

c_j		x					$c_j * x_j$	
2	x_1	18	26	3	1	5	2	
5	x_2	12	2	6	19	8	10	3
7	x_3	6	5	2	6	5	14	2
3	x_4	9	3	2	3	4	6	
8	x_5	37	8	**2.17**	**2.11**	20	16	
6	x_6	5	6	13	**3.8**	5	18 (**22.8**)	1
9	x_7	18	3	4	**2.11**	5	18(**18.9**9)	
4	x_8	7	4	1	3	20	4	
3	x_9	4	8	3	6	10	9	
11	x_{10}	37	13	1	2	40	11	

Arrangement of Promising variables $J = \{x_6, x_3, x_2\}$

		$x_6 = 3$	$x_3 = 2$
x_1	0	0	0
x_2	6	6	6
x_3	0	0	1
x_4	0	0	0
x_5	0	0	0
x_6	5	6	6
x_7	0	0	0
x_8	0	0	0
x_9	0	0	0
x_{10}	10	10	10

$$\qquad\qquad\qquad \downarrow \qquad \downarrow$$

$$P_0 = \begin{bmatrix} 37 \\ 26 \\ 13 \\ 19 \\ 40 \end{bmatrix} \qquad \begin{bmatrix} 30 \\ 22 \\ 12 \\ 14 \\ 32 \end{bmatrix} \qquad \begin{bmatrix} 24 \\ 17 \\ 6 \\ 11 \\ 25 \end{bmatrix}$$

Construction of θ matrix

c_j	x						$c_j * x_j$	
2	x_1	12	17	**1.5**	**1.1**	3	2	
5	x_2	8	1	3	11	5	5	3
7	x_3	4	3	1	3	3	7	2
3	x_4	6	2	1	2	2	3	
8	x_5	24	5	**1**	**1.2**	12	8	
6	x_6	3	4	6	2	3	12	1
9	x_7	12	2	2	1	3	9	
4	x_8	4	2	0	1	12	0	
3	x_9	2	5	1	3	6	3	
11	x_{10}	24	8	0	1	25	0	

Arrangement of Promising variables $J = \{x_6, x_3, x_2\}$

Construction of θ matrix

$$
\begin{array}{cccccc}
 & x_6 = 2 & c_j & x & & c_j * x_j \\
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{array}
\begin{bmatrix} 0 \\ 6 \\ 1 \\ 0 \\ 0 \\ 6 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix} &
\begin{bmatrix} 0 \\ 6 \\ 1 \\ 0 \\ 0 \\ 7 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix} &
\begin{array}{c} 2 \\ 5 \\ 7 \\ 3 \\ 8 \\ 6 \\ 9 \\ 4 \\ 3 \\ 11 \end{array} &
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{array}
\begin{bmatrix} 8 & 13 & 1 & 0 & 2 \\ 5 & 1 & 2 & 6 & 3 \\ 2 & 2 & 0 & 2 & 2 \\ 4 & \mathbf{1.86} & \mathbf{1} & \mathbf{1.2} & \mathbf{1.89} \\ 17 & 4 & 0 & 0 & 8 \\ 2 & 3 & 5 & 1 & 2 \\ 8 & 1 & 1 & 0 & 2 \\ 3 & 2 & 0 & 1 & 8 \\ \mathbf{1.89} & 4 & \mathbf{1.25} & 2 & 4 \\ 17 & 6 & 0 & 0 & 17 \end{bmatrix} &
\begin{array}{c} 0 \\ 5 \\ 0 \\ 3 \\ 0 \\ 6 \\ 0 \\ 0 \\ 3\,(\mathbf{3.75}) \\ 0 \end{array} &
\begin{array}{c} \\ 2 \\ \\ \\ \\ 1 \\ \\ \\ (3.75)3 \\ \\ \end{array} \\
\downarrow & \downarrow & & & & \\
P_0 = \begin{bmatrix} 24 \\ 17 \\ 6 \\ 11 \\ 25 \end{bmatrix} & \begin{bmatrix} 17 \\ 13 \\ 5 \\ 6 \\ 17 \end{bmatrix} & & & &
\end{array}
$$

Arrangement of Promising variables $J = \{x_6, x_2, x_9\}$

Construction of θ matrix

$$
\begin{array}{ccccccc}
 & x_6 = 1 & c_j & x & c_j * x_j & & x_2 = 1 \\
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{array}
\begin{bmatrix} 0 \\ 6 \\ 1 \\ 0 \\ 0 \\ 7 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix} &
\begin{bmatrix} 0 \\ 6 \\ 1 \\ 0 \\ 0 \\ 8 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix} &
\begin{array}{c} 2 \\ 5 \\ 7 \\ 3 \\ 8 \\ 6 \\ 9 \\ 4 \\ 3 \\ 11 \end{array} &
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{array}
\begin{bmatrix} 5 & 9 & 1 & 0 & 1 \\ 3 & \mathbf{1} & 2 & \mathbf{1} & \mathbf{1.8} \\ 1 & 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 1 \\ 10 & 3 & 0 & 0 & 4 \\ 1 & 2 & 4 & 0 & 1 \\ 5 & 1 & 1 & 0 & 1 \\ 2 & 1 & 0 & 0 & 4 \\ 1 & 3 & 1 & 0 & 2 \\ 10 & 4 & 0 & 0 & 9 \end{bmatrix} &
\begin{array}{c} 0 \\ 5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} &
\begin{array}{c} \\ 1 \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} &
\begin{bmatrix} 0 \\ 7 \\ 1 \\ 0 \\ 0 \\ 8 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix} \\
\downarrow & \downarrow & & & & & \downarrow \\
P_0 = \begin{bmatrix} 17 \\ 13 \\ 5 \\ 6 \\ 17 \end{bmatrix} & \begin{bmatrix} 10 \\ 9 \\ 4 \\ 1 \\ 9 \end{bmatrix} & & & & & \begin{bmatrix} 7 \\ 0 \\ 2 \\ 0 \\ 4 \end{bmatrix}
\end{array}
$$

Arrangement of Promising variables $J = \{x_2\}$

Solution

$x_2 = 7, x_3 = 1, x_6 = 8, x_{10} = 10 \quad \Rightarrow \quad Z = 200$

IV. LIMITATION

This algorithm has been tested with less than or equal to constraints and assumed the α value as 0.5.[9] Further the research is going on with mixed constraints.

V. CONCLUSION

In this paper an algorithm to solve integer linear programming problem is presented. This algorithm consists of two phases. A maximum of m promising variables are selected and arranged where m is the number of constraints. In phase two the arranged variables are allowed to enter into the basis with an integer value which is less than the maximum value it can assume. In the exiting method while finding the improved basic feasible solution it moves along the edge of the feasible region. In the proposed integer linear programming algorithm the improved solution moves in the interior of the feasible region. In the proposed method each time the variable is allowed to enter at αx_j value where α is less than 1. It has been assumed that α will take a value 0.5 and this assumption has to lead the optimum solution. But this has been proved further research is on to find out the optimum value of α for the fast convergence to the optimum solution.

VI. REFERENCES:

[1] Beale, E.M.L., "Mathematical Programming : Algorithms " Publications in Operations Research, No.16, 1969, pp 139-155.
[2] Dantzig, G.B. (1963) Linear Programming and Extensions, Princeton.
[3] Hamdy A Taha, Operations Research - An Introduction, Seventh Edition. PHI
[4] Julius S. Arorofsky, "Progress in Operations Research", Vol. III, Publications in Operations Research, No.16, John Wiley and Son Inc., 1969, pp 151-153.
[5] Karthikeyan G (2009). "Design Of A New Computer Oriented Algorithm To Solve Linear Integer Programming Problems", PhD Thesis, Alagappa University, India.
[6] Karthikeyan G and Sakthivel S, "Solution of Linear Integer Programming Problem", ICORAID-2005-ORSI, Bangalore, India.
[7] Nagarajan, S (2000) " A complex algorithm to solve large scale linear programming problems" Ph.D Thesis, Alagappa Univeristy, Karaikudi.
[8] Ragner Frisch, "The Multiplex method for Linear Programming". Memorandum of the University of Oslo. 1955, pp 1-6.
[9] Sakthivel, S (1984) "A multiplex algorithm to solve large scale linear programming problems " Ph.D Thesis,Anna University, Madras.

