Dhaka Univ. J. Sci. 61(2): 173-178, 2013 (July)

Additive Algorithm for Solving 0-1 Integer Linear Fractional Programming Problem

Md. Rajib Arefin, Touhid Hossain and Md. Ainul Islam*

Department of Mathematics, Dhaka University, Dhaka-1000, Bangladesh

(Received: 9 September 2012; Accepted: 6 March 2013)

Abstract

In this paper, we present additive algorithm for solving a class of 0-1 integer linear fractional programming problems (0-1 ILFP) where all the coefficients at the numerator of the objective function are of same sign. The process is analogous to the process of solving 0-1 integer linear programming (0-1 ILP) problem but the condition of fathoming the partial feasible solution is different from that of 0-1 ILP. The procedure has been illustrated by two examples.

Key words: 0-1 Integer programming, Dual feasibility, Implicit enumeration.

I. Introduction

An integer programming where all the variables are restricted to a value of 0 or 1 is called 0-1 integer programming or binary programming. Binary variables bear immense importance because they occur regularly in many model formulations, such as set covering problem, assignment problem, traveling salesperson problem, capital budgeting problem, knapsack problem, timetabling problem etc. Several methods have been developed for solving 0-1 integer programming problems. Techniques available for solving the 0-1 integer programming problem include algorithms of Glass[7], Balas[3], Glover[8], Lawler and Bell[10], Geoffrion[6], Lemke and Spielberg[11] etc. These algorithms are enumerative and developed for solving zero-one Integer Linear Programs (ILP). But often we may have to deal with some problem where the objective function is the ratio of two linear functions called Linear Fractional Programming (LFP) problem. A Linear Programming (LP) computes a policy delivering the best outcome, such as maximum profit or lowest cost whereas a LFP is used to achieve the highest ratio of outcome to cost; the ratio represents the highest efficiency. Therefore, LFP is sometimes more informative than LP. Our purpose is to solve such types of problem using an enumerative algorithm. In this paper, we have used additive algorithm of Balas[3] for solving a class of 0-1 ILFP problems. Balas developed this algorithm for solving 0-1 integer linear programs. Puri and Swarup[12] suggested the extreme point mathematical programming technique for solving 0-1 ILP as well as ILFP. The extreme point mathematical programming is mainly based on simplex algorithm. But the additive algorithm does not require the simplex algorithm. The general idea of additive algorithm is to enumerate (implicitly) all 2^n possible solutions of a problem. The first requirement of this algorithm is dual feasibility. Initially, it is easier to obtain the dual feasibility of any LP problem but it is difficult for LFP problem. The main challenge in LFP is to obtain the dual feasibility at the initial stage. In this paper, we consider a class of 0-1 ILFP as follows:

$$\text{Minimize } Q(x) = \frac{\sum_{j=1}^{n} p_j x_j}{\sum_{j=1}^{n} d_j x_j + d_0}, \qquad (1.1)$$

(all p_j are of same sign)

Subject to

$$\sum_{j=1}^{n} a_{ij} x_j (\le, \ge) b_i \,, i = 1,2,\dots,m \qquad (1.2)$$

$$x_j = 0 \text{ or } 1 \text{ for all } j \qquad (1.3)$$

In the Section III, theoretically we have shown that, dual feasibility can be obtained from (1.1)-(1.3). Many real life oriented models can be formulated with this type of structure such as set covering problem[1]. In the next Section, we summarize the idea of additive algorithm.

II. Summary of Additive Algorithm for Solving 0-1 ILP Developed by E. Balas

For the purpose of this algorithm, the continuous version of the 0-1 problem must start with the dual-feasibility, that is, optimal but not feasible. Moreover, all the constraints must be of $\le$ type. This format can be achieved as follows:

Let the problem be of the minimization type (there is no loss of generality here) and define it as

$$\text{Minimize } z = \sum_{j=1}^{n} c_j x_j \,, \text{ all } c_j \ge 0 \qquad (2.1)$$

Subject to

$$\sum_{j=1}^{n} a_{ij} x_j + S_i = b_i \,, i = 1,2,\dots,m \qquad (2.2)$$

$x_j = 0$ or 1 for all j

$S_i \ge 0$, for all i.

Where, S_i is the slack variable associated with the ith constraint. The continuous version of foregoing problem is dual feasible if every $c_j \ge 0$. Any $c_j < 0$ can be converted to the desired format by complementing the variable x_j, that is, by substituting $x_j = 1 - x_j'$, where x_j' is a binary variable, in the objective function and constraints. If in addition to dual feasibility, the problem is primal-feasible, nothing

*Author for Correspondence, *e-mail*: mainul_51@yahoo.com more need to be done, since the minimum, in terms of new variables is achieved by assigning zero values to all the variables. However, if it is primal infeasible, the additive algorithm is used to find the optimum. The general idea of the additive algorithm is to enumerate all 2^n possible solutions of the problem. However, it recognizes that some solutions can be discarded automatically without being investigated implicitly. Hence, in the final analysis, only a portion of the 2^n solutions need to be investigated explicitly.

In terms of the given zero-one problem (2.1)-(2.2), this idea is implemented as follows: Initially, assume that all the variables are at zero level. This is logical, since all $c_j \geq 0$. Since the corresponding solution is not feasible (i.e. some slack variables S_i may be negative), it will be necessary to elevate some variables to level one. The procedure calls for elevating one (or perhaps more) variable at a time, provided there is evidence that this step will be moving the solution toward feasibility, that is, making $S_i \geq 0$ for all i. Balas[3] developed a number of tests to ensure the proper selection of the variables to be elevated to level one. The following terms are needed to describe these tests.

Free variable: A free variable is initially at zero level but may be elevated to level one if this can improve the infeasibility of the problem.

Partial solution: A partial solution provides a specific binary assignment for some of the variables in the sense that it fixes the values of one or more variables at zero or one. A convenient way to summarize this information for the purpose of the (Branch and Bound) algorithm is to express the partial solution as an ordered set. Let J_t represent the partial solution at the tth node (or iteration), and let the notation $+j(-j)$ represent $x_j = 1$ $(x_j = 0)$. Thus the elements of J_t consist of the subscripts of the fixed variables with the plus (minus) sign signifying that the variable is one (zero). The set J_t must be ordered in the sense that each new element is always augmented on the right of the partial solution.

A partial solution is said to be fathomed if

1. It cannot lead to a better value of the objective function.
2. It cannot lead to a feasible solution.

When all the elements of a fathomed partial solution are negative, the associated variables would have been considered at both zero and one levels. As a result, there are no more branches to consider and the enumeration is complete. The general version of the additive algorithm is now presented by the concept of partial solutions. The exclusion tests used to fathom partial solutions and augment new variables at level one are also generalized for the zero-one problem.

Consider the binary problem (2.1)-(2.2). Let J_t be the partial solution at node t (initially, $J_0 = \emptyset$, which means that all variables are free) and assume z^t is the associated value of z while $\bar{z}$ is the current best upper bound (initially $\bar{z} = \infty$). Balas [3] developed the following four tests to ensure the proper selection of the variables to be elevated to level one.

Test 1: For any free variable x_r, if $a_{ir} \geq 0$ for all i corresponding to $S_i^t < 0$, then x_r cannot improve the infeasibility of the problem and must be discarded as non promising.

Test 2: For any free variable x_r, if $c_r + z^t \geq \bar{z}$ then x_r cannot lead to an improved solution and hence must be discarded.

Test 3: Consider the ith constraint $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + S_i = b_i$, for which $S_i^t < 0$. Let N_t define the set of free variables not discarded by test 1 and 2. None of the free variable in N_t are promising if for at least one $S_i^t < 0$, the following condition is satisfied: $\sum_{j \in N_t} \min\{0, a_{ij}\} > S_i^t$. This actually says that the set N_t cannot lead to a feasible solution and hence must be discarded altogether. In this case, J_t is said to be fathomed.

Test 4: If $N_t \neq \emptyset$, the branching variable x_k is selected as the one corresponding to

$$v_k^t = \max\{v_j^t\}, j \in N_t,$$

where, $$v_j^t = \sum_{i=1}^{m} \min\{0, S_i^t - a_{ij}\}.$$

If $v_k^t = 0, x_k = 1$ together with J_t yields an improved feasible solution. In this case, J_{t+1}, which is defined by J_t with $\{k\}$ augmented on the right, is fathomed. Otherwise, the foregoing tests are applied again to J_{t+1} until the enumeration is completed, that is, until all the elements of the fathomed partial solution are negative.

In the following section, we generalize the additive algorithm for solving a 0-1 ILFP.

III. Additive Algorithm for Solving 0-1 ILFP

Before using the additive algorithm for solving a 0-1 ILFP, we have to ensure the dual feasibility of that problem. In this section, theoretically it has been shown that dual feasibility can be obtained from the problem (1.1)-(1.3). Introducing slack variables we obtain the following from (1.1)-(1.3):

$$\text{Minimize } Q(x) = \frac{\sum_{j=1}^{n} p_j x_j}{\sum_{j=1}^{n} d_j x_j + d_0}, \qquad (3.1)$$

(all p_j are of same sign)

Subject to

$$\sum_{j=1}^{n} a_{ij} x_j + S_i = b_i, \; i = 1,2,\ldots,m \qquad (3.2)$$

$$x_j = 0 \text{ or } 1 \text{ for all } j \qquad (3.3)$$

Where $S_i, i = 1,2,\ldots,m$ are the slack variables and $S_i \geq 0$, for all i.

This is a minimization problem, so at the initial tableau all of the reduced cost factors (i.e. Δ_j) should be ≥ 0 to obtain the dual feasibility. From (3.1) and (3.2) it is clear that initially $x_j = 0$, for $j = 1,2,\dots,n$. Thus $Q(x)$ becomes zero. Thus, from Bajalinov[2] we obtain the reduced cost factor Δ_j of the problem (3.1)-(3.2) as;

$$\Delta_j(x) = (\Delta'_j - Q(x)\Delta''_j) \qquad (3.4)$$

Where $\Delta'_j = p_j - \sum_{i=1}^{m} p_{s_i} x_{ij}$,

$\Delta''_j = d_j - \sum_{i=1}^{m} d_{s_i} x_{ij}$. Δ'_j and Δ''_j are referred to as reduced cost of numerator and denominator respectively. Equation (3.4) yields that

$$\Delta_j(x) = \Delta'_j - 0, \text{since } Q(x) = 0,$$

$$\text{i.e. } \Delta_j(x) = p_j - \sum_{i=1}^{m} p_{s_i} x_{ij} \qquad (3.5)$$

where, p_{s_i} are the coefficients of the basic variables at the numerator of the objective function (3.1). Since $p_{s_i} = 0$ at the initial stage of the problem (3.1)-(3.3) for all i, we obtain from (3.5) that $\Delta_j = p_j,\ j = 1,2,\dots,n$. It was assumed in (3.1) that $p_j \geq 0$ for all j.

So $\Delta_j(x) \geq 0$ for all j. Therefore, the dual feasibility of the problem (3.1)-(3.3) is guaranteed.

Once dual feasibility of the problem (3.1)-(3.3) is guaranteed, additive algorithm can be used to obtain the optimal solution of the problem (1.1)-(1.3). That is, the four tests of the Section II can be incorporated to the problem (3.1)-(3.3). The condition for fathoming a partial feasible solution for 0-1 ILFP is given in the next section.

IV. Condition of Fathoming Partial Feasible Solution for 0-1 ILFP

It should be noted here that the condition for fathoming a partial feasible solution in 0-1 ILFP is different from the case of 0-1 ILP. In case of 0-1 ILP, a partial solution is fathomed if it yields a feasible solution, because elevation of another variable to level one together with current partial feasible solution gives the value of the objective function greater than the current value of the objective function (minimization problem). On the other hand, a partial feasible solution of 0-1 ILFP does not guarantee that the solution can be fathomed, since elevation of another variable to level one together with the current feasible partial solution may increase or decrease the value of the objective function with satisfying all the constraints. The objective function in 0-1 ILFP is the ratio of two linear functions, so if a variable is elevated to level one it may increase or decrease the value of the objective function because of the effect of its coefficients at the numerator and denominator. This is why, fathoming a partial feasible solution in 0-1 ILFP is different from the case of 0-1 ILP. This difference has been discussed in iteration 3 of example II.

Now let us illustrate the above procedure by two examples.

Example I:

$$\text{Minimize } Q(x) = \frac{2x_1 + x_2 + 3x_3 + x_4}{x_1 - 2x_2 + x_3 - x_4 + 6}$$

Subject to

$$2x_1 + 2x_2 - x_3 - x_4 \leq 2$$

$$x_1 - 2x_2 + 3x_3 + x_4 \geq 1$$

$$x_1 + x_2 + x_3 - x_4 \leq 3$$

$$x_1, x_2, x_3, x_4 \ \text{ are binary}.$$

Since the second constraint is " $\geq$ " type, multiply the constraint by -1. Then introducing slack variables S_1S_2, S_3 we get

the following table 1:

Table 1. Initial tableau for Example I

		P	2	1	3	1	0	0	0	
		D	1	-2	1	-1	0	0	0	b
D_B	P_B	Basis↓	x_1	x_2	x_3	x_4	S_1	S_2	S_3	
0	0	S_1	2	2	-1	-1	1	0	0	2
0	0	S_2	-1	2	-3	-1	0	1	0	-1
0	0	S_3	3	1	1	-1	0	0	1	3
		Δ'_j	2	1	3	1	0	0	0	
		Δ''_j	1	-2	1	-1	0	0	0	Q = 0
		$\Delta_j = \Delta'_j - Q\Delta''_j$	2	1	3	1	0	0	0	

Since all the Δ_j's are non negative and b_2 is negative so the problem is dual feasible.

Iteration 0: For $J_0 = \emptyset,\ \bar{Q} = \infty,$

$(S_1^0, S_2^0, S_3^0) = (2, -1, 3)$

Since the constraint coefficient of x_2 corresponding to the negative slack variable is nonnegative, x_2 is excluded by test 1. By test 3, $N_0 = \{1,3,4\}$ cannot be abandoned because

$S_2: -1-3-1 = -4 < -1$

By test 4, $v_1^0 = 0+0+0 = 0;$

$v_3^0 = 0 + (-1-2) + 0 = -3;$

$v_4^0 = 0+0+0 = 0$

$\max v_j^0 = 0,\ j \in N_0$. This is a tie, choose $k = 4$, since x_4 yields the minimum value of $Q(x)$.

Iteration 1: For $J_1 = \{4\},\ \bar{Q} = \frac{1}{5},$

$$(S_1^1, S_2^1, S_3^1) = (2+1, -1+1, 3+1)$$

$$= (3,0,4) \quad Q^1 = 1/5.$$

Since it is feasible, $\bar{Q} = Q^1 = 1/5$. No

other variables are promising to improve the current value of the objective function so J_1 is fathomed.

Iteration 2: For $J_2 = \{-4\},\ \bar{Q} = 1/5,$

$(S_1^2, S_2^2, S_3^2) = (2,-1,3),\ Q^2 = 0$

If we elevate x_1 to level one then we will obtain $Q = 2/7$, which is greater than $\bar{Q}$ (upper bound of Q). So, by test 2, discard x_1. Similarly substitution of $x_3 = 1$ would give the value of Q greater than $\bar{Q}$. Thus test 2 also excludes x_3. Hence J_2 is fathomed. Since the element of J_2 is negative, the enumeration is complete and J_1 is optimal. Therefore, the solution of the given example is $(0,0,0,1)$ with $\min Q = 1/5$.

We now present a real life example which illustrates the application of the method. The problem has been formulated using the model of set covering problem (SCP) for LFP[1]. We have used the Section II and III to solve this problem.

Example II: Installing Fire Stations at Different Cities

Suppose there are six cities (cities 1-6) in the district. The district is reviewing the location of its fire stations. In each city the investments required to build fire stations differ from one another. The investments required in each city are shown in table 2. The district must determine where to build fire stations. The fire station can be placed in any city but the district wants to build the stations such that at least one fire station be within 15 minutes (driving time) of each city and the total cost invested per number of stations built be minimal. The times (in minutes) required to drive between the cities in the district are shown in table 3.

For each city we must define one variable

$x_j = 0$ or 1, $j = 1,2,\dots,6$. This variable x_j

will be 1 if we place a station in city j, and

will be 0 otherwise. Then the problem can be

formulated as follows:

$$\text{Miniimize} \quad Q(x) = \frac{12x_1 + 14x_2 + 9x_3 + 16x_4 + 5x_5 + 8x_6}{x_1 + x_2 + x_3 + x_4 + x_5 + x_6}$$

Subject to

$$\begin{aligned}
x_1 + x_2 \qquad\qquad\qquad &\geq 1 \\
x_1 + x_2 \qquad\qquad + x_6 &\geq 1 \\
x_3 + x_4 \qquad\qquad &\geq 1 \\
x_3 + x_4 + x_5 \qquad &\geq 1 \\
x_4 + x_5 + x_6 &\geq 1 \\
x_2 \qquad + x_5 + x_6 &\geq 1 \\
x_j = 0 \text{ or } 1,\ j = 1,2,3,\dots,6 &
\end{aligned}$$

Table 2. Investment required in the city

	City 1	City 2	City 3	City 4	City 5	City 6
Millions(taka)	12	14	9	16	5	8

Table 3. Driving time in minutes

	City 1	City 2	City 3	City 4	City 5	City 6
City 1	0	10	20	30	30	20
City 2	10	0	25	35	20	10
City 3	20	25	0	15	30	20
City 4	30	35	15	0	15	25
City 5	30	20	30	15	0	14
City 6	20	10	20	25	14	0

As required in additive algorithm that all the constraints must be less than or equal ($\leq$) form. The constraints of the given SCP problem must be converted to '$\leq$' form to start with the dual feasibility. To do this, multiply each constraint by -1. Also all the coefficients at the numerator of the objective function is nonnegative. Then introducing slack variables $(S_1, S_2, S_3, S_4, S_5, S_6)$ to the constraints, we get the following problem which is dual feasible at the initial stage:

$$\text{Minimize} \quad Q(x) = \frac{12x_1 + 14x_2 + 9x_3 + 16x_4 + 5x_5 + 8x_6}{x_1 + x_2 + x_3 + x_4 + x_5 + x_6}$$

Subject to

$$-x_1 - x_2 + S_1 = -1$$

$$-x_1 - x_2 - x_6 + S_2 = -1$$

$$-x_3 - x_4 + \ S_3 = -1$$

$$-x_3 - x_4 - x_5 + S_4 = -1$$

$$-x_4 - x_5 - x_6 + S_5 = -1$$

$$-x_2 - x_5 - x_6 + S_6 = -1$$

$$x_j = 0 \text{ or } 1,\ j = 1,2,3,\dots,6$$

Now let us apply the procedure for solving 0-1 ILFP proposed in Section III.

Iteration 0: For $J_0 = \emptyset, \overline{Q} = \infty$,

($\overline{Q}$ represents the current best upper bound of the objective function)

$(S_1^0, S_2^0, S_3^0, S_4^0, S_5^0, S_6^0) = (-1,-1,-1,-1,-1,-1),\ Q^0 = 0$

By test 3, $N_0 = \{1,2,3,4,5,6\}$ cannot be abandoned because,

$S_1 :\ -1 - 1 = -2 < -1;$

$S_2 :\ -1 - 1 - 1 = -3 < -1;$

$S_3 :\ -1 - 1 = -2 < -1;$

$S_4 :\ -1 - 1 - 1 = -3 < -1;$

$S_5 :\ -1 - 1 - 1 = -3 < -1;$

$S_6 :\ -1 - 1 - 1 = -3 < -1$

By test 4, $v_1^0 = -4, v_2^0 = -3,\ v_3^0 = -4,\ v_4^0 = -3,\ v_5^0 = -3,\ v_6^0 = -3$

Now $\max v_j^0 = -3,\quad j \in N_0$ which occurs for $v_2^0,\ v_4^0\ \&\ v_5^0,\ v_6^0$. Choose $k = 5$ since, $x_5 = 1$ gives the minimum value of $Q(x)$ among the variables x_2, x_4, x_5, x_6.

Iteration 1: For $J_1 = \{5\}, \overline{Q} = \infty$,

$(S_1^1, S_2^1, S_3^1, S_4^1, S_5^1, S_6^1) = (-1,-1,-1,0,0,0)$

$Q^1\ = 5$

By test 3, $N_1 = \{1,2,3,4,6\}$ cannot be abandoned because in this case,

$S_1 :\ -2 < -1;\ S_2 : = -3 < -1;$

$S_3 : = -2 < -1;\ S_4 : = -2 < 0;$

$S_5 : = -2 < 0;\ S_6 : = -2 < 0.$

By test 4,

$v_1^1 = -1, v_2^1 = -1, v_3^1 = -2, v_4^1\ - 2,$

$v_6^1 = -2.$

Now $\max v_j^1 = -1,\ j \in N_1$ which occurs for $v_1^1 \& v_2^1$. Choose $k = 1$ since $x_1 = 1$ yields the minimum value of $Q(x)$.

Iteration 2: For $J_2 = \{5,1\},\ \overline{Q} = \infty$,

$(S_1^2, S_2^2, S_3^2, S_4^2, S_5^2, S_6^2)$

$= (-1 + 1, -1 + 1, -1,0,0,0)$

$= (0,0,-1,0,0,0),\ Q^2 = 17/2.$ By test 3, $N_2 = \{2,3,4,6\}$ cannot be abandoned.

By test 4, $v_2^2 = -1, v_3^2 = 0, v_4^2 = 0$, and

$v_6^2 = -1.$

Now $\max v_j^2 = 0,\ j \in N_2$ which occurs for $v_3^2 \& v_4^2$. Choose $k = 3$ since $x_3 = 1$ yields the minimum value of $Q(x)$.

Iteration 3: For $J_3 = \{5,1,3\}$,

$(S_1^3, S_2^3, S_3^3, S_4^3, S_5^3, S_6^3) = (0,0,0,0,0,0),$

$Q^3 = 26/3$

This is feasible. But, this partial feasible solution may not be fathomed because elevation of some other variables to level one may give the value of Q less than Q^3 satisfying all the constraints. Now we look for those variables.

a) Elevation of x_2 to level one yields $Q = 10$, which is greater than Q^3

b) Elevation of x_4 to level one yields $Q = 42/4$, which is greater than Q^3

c) Elevation of x_6 to level one yields $Q = 17/2$, which is less than Q^3

Therefore, x_6 can be elevated to level one since it gives the value of Q less than Q^3. Now test whether elevation x_6 together with the partial solution J_3 satisfies all the constraints.

$$(S_1^3, S_2^3, S_3^3, S_4^3, S_5^3, S_6^3)$$

$$= (0,0 + 1,0,0,0 + 1,0 + 1)$$

$$= (0,1,0,0,0,1,1)$$

This is still feasible. Thus, select x_6 to level one and rewrite the partial solution J_3 as

$J_3 = \{5,1,3,6\}$ with $\overline{Q} = 17/2$. Hence, J_3 is fathomed.

Iteration 4: For $J_4 = \{5,1,-3,-6\}, \overline{Q} = \frac{17}{2}$,

$(S_1^4, S_2^4, S_3^4, S_4^4, S_5^4, S_6^4) = (0,0,-1,0,0,0)$,

$Q^4 = 17/2$.

Now by test 2, we discard x_2 and x_4 because elevation of x_2 or x_4 to level one yields the value of Q greater than $\overline{Q} = 17/2$. So $N_4 = \emptyset$ and hence, J_4 is fathomed.

Iteration 5: For $J_5 = \{5,-1,-3,-6\}$,

$\overline{Q} = \dfrac{17}{2}, (S_1^5, S_2^5, S_3^5, S_4^5, S_5^5, S_6^5)$

$= (0,0,-1,0,0,0), Q^5 = 5$.

Test 2 discards x_2 and x_4 again. So $N_5 = \emptyset$ and hence, J_5 is fathomed.

Iteration 6: For $J_6 = \{-5,-1,-3,-6\}$, $\overline{Q} = 17/2$.

Clearly, this partial is not promising any more. Therefore, J_6 is fathomed. Since all the elements of J_6 are negative, the enumeration is complete and J_3 is optimal.

Thus, the solution of the given SCP is (1, 0, 1, 0, 1, 1) with $\min Q = 17/2$.

V. Conclusion

In this paper, the additive algorithm has been used for solving a class of 0-1 ILFP where all the coefficients p_j at the numerator of the objective function are of same sign. The process is easy to understand and doesn't require the simplex algorithm. Future work should extend the process for solving any kind of 0-1 ILFP using the additive algorithm.

References

1. Arora, S.R., M.C. Puri, K. Swarup, 1977. The Set Covering Problem with Linear Fractional Functional. *Indian Journal of Pure and Applied Mathematics*, ***8(5)***, *578-588*.
2. Bajalinov, E. B., 2003. Linear Fractional Programming: Theory, Methods, Applications and Software. *Kluwer Academic Publishers, Dordrecht, the Netherlands.*
3. Balas, E. 1965. An Additive Algorithm for Solving Linear Programs with Zero-One variables. *Operations Research*, ***13(4)***, *517-546*.
4. Balas, E. and E. Zemel, 1980. An algorithm for Large 0-1 Knapsack Problems. *Operations Research,* ***28(5)***, *1130–1154.*
5. Daskalaki, S., Birbas T. and Housos 2004. E. An Integer Programming Formulation for a Case Study in University Timetabling. *European Journal of Operational Research,* ***153(1)***, *117–135.*
6. Geoffrion, Arthur M., 1966. Integer Programming by Implicit Enumeration and Balas' Method. *SIAM Rev.,* ***9(2)***, *178–190.*
7. Glass, Harvey, 1965. A Zero-One Algorithm". *Honeywell Aeronaut. Div., Petersburg, Fla*.
8. Glover, Fred 1965. A multiphase-dual algorithm for the zero-one integer programming problem. *Operations Research,**13(6)**, 879-919.*
9. Hamdy, A., Taha, 2007. Operations Research: An Introduction, 8th ed. *Pearson Education, Inc.*
10. Lawler, E. L., and M. D. Bell, 1966. A Method for Solving discrete optimization problems. *Operations Research,* ***14(6)***, *1098-1112*.
11. Lemke, C. E., and K. Spielberg, 1967. Direct Search Zero-One and Mixed Integer Programming. *Operations Research,* ***15(5)***, *892-914*.
12. Puri, M. C., K. Swarup, 1974. "Extreme Point Linear Fractional Functional Programming, *Z. O. R.,* ***18(3)***, *131-139.*