An Implicit Enumeration Algorithm for Mixed-Integer-Linear Programming

P.Radhakrishnan
Faculty of Information Science and Technology
Multimedia University, Bukit Beruang, Melaka, 75450, Malaysia
Email: radha.krishnan@mmu.edu.my
M.V.C.Rao
Faculty of Engineering and Technology
Multimedia University, Bukit Beruang, Melaka, 75450, Malaysia
Email: machavaram.venkata@mmu.edu.my

Abstract

An endeavour is made in this paper to present a novel implicit branch and bound method for mixed integer linear programming problems. The integer variables are driven to their non-basic variables, which can be fixed, at either lower or upper bound, remaining non-basic variables as such. The concept of twin models is exploited to get more number of nodes to get the optimal solution. This method eliminates non-optimal solutions and thus reduces the number of iterations. A different and simple method of adding a new constraint is formulated. Many typical test problems are solved on a micro computer to highlight the efficacy of this method. The accuracy obtained was good. The fact that it required many branching to solve even a small model is a strong evidence of its invulnerability to round off error. Its memory requirement is also modest. It is highly effective in solving a very large class of integer programming models, mixed integer, pure integer and the special class involving only binary variables. This does not add any new constraints to the original constraint set, since it merely manipulates upper and lower bounds on the integer variables. Hence, it can be said that it is immune to round-off errors. Experience indicates that the use of surrogate constraint is effective in improving the computation time. The solution time varies almost exponentially with the number of variables. So problems up to 100 variables can be solved in a reasonable amount of computation time. The most important advantages are its immunity to round off errors and its modest memory requirement in comparison with other methods like branch and bound and cutting plane method.

Introduction

Branch and Bound approach is the most widely used one for solving all-integer and mixed-integer-programming models, but it is well known that B&B suffers from round-off error, and in many cases requires substantial computer storage resources. In implicit enumeration algorithm, both the disadvantages are eliminated.

MILP is an important tool for modeling and analysis, but finding solutions to even modest-sized MILP models without resorting to scarce and expensive 'Super-computing' is challenging and problematic at best. *Land and Dong* developed an approach called branch-and-bound that begins with the optimal solution of the LP relaxation, and generates explicit lower-and upper-bound constraints on integer decision variables, whose relaxed solution values are fractional. Iterations proceed in a tree like fashion, appending these 'cuts' until an integer solution is located. Two serious difficulties with B&B are the effects of round-off error and the requirement in many cases for massive amounts of computer storage. The former can result in (many cases) sub-optimal solutions, with no clue to the analyst that this has happened. The latter may become exceedingly expensive in terms of computer resources with no forewarning.

The implicit enumeration algorithm starts out with all variables set equal to 0 and then systematically specifies certain variables to take on the value 1 until a feasible solution is obtained. This first feasible solution is then considered to be the best feasible solution to date. Since the variables are chosen heuristically, it is quite possible that the first feasible solution may not be optimal. Consequently, the algorithm systematically looks at various combinations of the variables set equal to 0 and 1 that can possibly improve on the best feasible solution, until an optimal solution is obtained. Many combinations that cannot possibly lead to a better feasible solution are not examined and are thus said to be implicitly enumerated. The details of how variables are chosen to be 0 or 1 and how blocks of possible solutions can be eliminated without explicitly enumerating them are quite lengthy, so first how the algorithm works is illustrated followed by the complete description of the algorithm.

Problem formulation and Notation

Consider the mixed integer-programming problem with the usual notations

Maximize $Z = C^T X$

Subject to $A.X = b$

$X \geq 0$ and X_j integers for $j \in \{I\}$ …(1)

The constraints from (1) are $B.X_B + D.X_D = b$. If, B^{-1} is the inverse of the basis, then the solution to the LP relaxation of (1) is $X_B = B^{-1}.b$, $X_D = 0$ Which is a basic solution; and If, $X_B \geq 0$, it is also a feasible solution.

Suppose the optimal solution to (1) includes $X_j = d_j$. $j \in \{I\}$, where d_j is an integer. Let us introduce an integer upper bound $U_j = d_j$ or lower bound $L_j = d_j$ and make the linear transformation. $X_j' = U_j - X_j$ If, dj is an upper bound $X_j' = X_j - L_j$ If, dj is a lower bound. Thus we get an optimal solution in which $X_j' = 0$.

Two linear programming models are said to be twin models if, they differ only in one or more integer bounds. The following model is a twin model of (1)

Maximize $Z = C^T X$, Subject to $A.X = b$

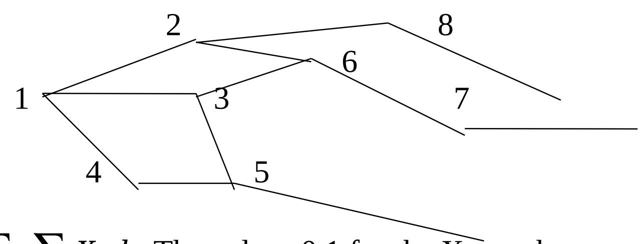
$X_j \geq 0$ for all j $U_j \geq X_j \geq L_j$ $j \in \{I\}$ …(2)

The Twin models (1) and (2) will prove useful in the exposition to follow. When we follow the twin model we get more no. of nodes to get the optimal solution. This method uses more computer storage than search methods do, but the best solution found is an optimal solution.

Suppose a network consists of N nodes 1,2,3,N and suppose we wish to find the shortest route from node 1 to node N, by way of the linked nodes. Suppose the distance from node i to node j is d_{ij} if, these two nodes are not linked then d_{ij} can be taken to be infinite. To find the shortest route from 1 to N we proceed as follows. Let $X_{ij}=1$ if, the branch from i to j is on the route; Otherwise $X_{ij}=0$.

Then for each I, on the route or not $\sum_j X_{ij} \leq 1$ for i =1 and for each j, on the route or not $\sum_j X_{ij} \leq 1$ for j = N. If, we arrive at a point we must leave it, so for each j, except 1 and N.

$$\sum_i X_{ij} = \sum_k X_{jk}$$



D = $\sum_{i} \sum_{j} X_{ij} d_{ij}$ The values 0,1 for the X_{ij} can be ensured by constraint.

The Implicit Enumeration Algorithm (IEA)

In this section we outline the IEA, suppressing the mathematical details for clarity of exposition

V = A list variable bounds for twin models

Z^* = Current best feasible solution to (1)

Z = Optimal solution to the LP relaxation of the current twin model.

Initially, V is empty. If, no explicit lower or upper bounds are available, set L_j=0 and U_j= +∞ for j∈{I}; set Z^* = -∞. Since the models are maximization ones, relax the integer restriction on X_j, j∈{I} in (2).

Algorithm

Step 1: Solve the resulting Linear programming model using a Revised Simplex Method. At the k^{th} iteration the solution is

$$X_B^{(k)} = B^{-1}.b, \qquad X_D^{(k)} = d^{(k)}$$

here $d^{(k)}$ is an integer vector of d_j (Upper and Lower bounds). If, there is no feasible solution, or if, $Z<Z^*$ go to Step 3. Otherwise, go to Step 2.

Step 2: At least one X_j∈{I}, is non-integer. Select the first non-integer X_j, j∈{I}. Generate the twin models as follows. Record the lower bound U_j=[X_j] so that X_j is required to be an element of { U_j old upper bound for X_j} and push this model into the V list.

Record the upper bound L_j =[X_j]+1 so that X_j is required to be an element of {old lower bound for X_j,L_j } and push this twin model onto the V list. The algorithm always begins with the previous optimal result. ([.] Denotes the greatest integer part). Go to Step 3.

Step 3: If, V is empty, go to Step 4. Otherwise, select the most recent bound in V and solve the resulting model (k+1) based on twin model (k) as follows:

$$X_B^{(k+1)} = X_B^{(k)} - A^{(k)}.(L^{(k+1)} - d^{(k)})$$
$$= B^{-1}.\{b - A(L^{(k+1)} - d^{(k)}) \qquad \text{Where } A^{(k)} = B^{-1}.A \text{ and } X_D^{(k+1)} = L_D^{(k+1)}$$

Set k = k+1 Go to Step 1.

Step 4: If, Z^* = -∞, there is no feasible solution for (1). If, Z^*>-∞ the feasible solution that yielded Z^* is optimal.

It is an improvement over the conventional simplex method. Because it does no tableau pivoting or basis-matrix inverse B^{-1} updating. The modified algorithm accommodates integer variables by use of a bounded algorithm together with a Branch and Bound strategy in which each successive model begins at a starting point that was the previous optimal solution. Through the use of upper-bounded LP concepts, the solution of (2) proceeds rapidly when it starts from the known solution to the twin model (1).

After using the recursion formula to obtain $C^T = C^T_B B^{-1}$ choose at the j^{th} iteration the entering variable that satisfies $(C_j - C^T a_j) > 0$ and $X_j=L_j$, $(C_j - C^T a_j) < 0$ and $X_j=U_j$
here a_j is the j^{th} column of matrix A. Subsequently, we will denote the updated current value as a'j $= B^{-1}a_j$, b' $= B^{-1}b$ etc.

The bounded technique used herein involves revision to the sub-algorithms for selection of the existing variable and for performing updates (adding elementary vectors to the string, etc.). This approach involves modification of the upper-bounded technique, since their will always is both an upper bound and a non-zero lower bound resulting from the branching step. The Implicit enumerated procedure eliminates non-optimal solutions and thus reduces the amount of calculation.

Procedure

- Find an initial feasible integer solution
- Branch: Select a variable and divide the possible solutions into two groups. Select one branch for investigation.
- Find the upper bound or maximum value for the problem defined by the branch selected. This bound can be found by considering the problem as a linear programming problem.
- Compare: Compare the bound obtained for the branch being considered with the best solution so far for the branches examined. If, the bound is less, delete the whole new branch. If, the bound is greater and an integer it becomes the new best solution so far. If, the bound is greater but not an integer, continue in this same branch by branching further.
- Completion: When all branches have been examined, the best solution so far is the optimal solution.

Geometrical Interpretation

Geometrical interpretation of implicit branch and bound method is the best explained through an illustration. Using IA method to solve the following IPP.

Maximize $Z = X_1+X_2$ Subject to $3X_1+2X_2\le12$, $X_2\le2$, X_1, $X_2\ge0$ and are integers.

The above problem can be rewrite as follows

Minimum $Z = -X_1-X_2$, Subject to $3X_1+2X_2+X_3=12$, X2 $X_4=2$, X1, $X_2\ge0$

The Optimal Solution of the above linear programming problem is $X_1=8/3$, and $X_2=2$ with Min Z=14/3. As this solution is not integer valued, the given linear programming problem is partitioned into two sub-problems. Since $X_1=8/3$ gives $2< 8/3 < 3$, the sub problems are

Sub problem 1	Sub problem 2
Maximize $Z = X_1+X_2$	Maximize $Z = X_1+X_2$
Subject to $3X_1+2X_2\le12$	Subject to $3X_1+2X_2\le12$
$X_2\le2$, $X_1\le2$	$X_2\le2$, $X_1\ge3$
X_1, $X_2\ge0$ and are integers.	X_1, $X_2\ge0$ and are integers.

The Optimal Solution of problem 2 is $X_1=2$, $X_2=2$ with max Z=4. Whereas the Optimal Solution of problem 3 is $X_1=3$ and $X_2 = 3/2$ with max Z=9/2. In sub problem 1, Since all the variables are integers there is no need to branch this problem further. But in sub problem 2 since X_2 is still non-integer it needs further subdivision.

Now since $X_2=3/2$ gives $1 < X_2 < 2$, we form two new sub problems by adding the constraints one by one in problem 3. The two additional sub-problems are

Sub problem 3

MaximizeZ=X_1+X_2, Subject to $3X_1+2X_2\le12$,$X_2\le2$,$X_1\ge3$,$X_2\le1$,X_1, $X_2\ge0$ and are integers.

Sub problem 4

Maximize$Z=X_1+X_2$, Subject to$3X_1+2X_2\le 12$, $X_2\le 2$, $X_1\ge 3$, $X_2\ge 2$,X_1,$X_2\ge 0$and are integers.

In sub problem 3 the constraint $X_2\le 2$ is redundant. The Optimal Solution to this problem is obtained as $X_1=10/3$ and $X_2=1$ with max $Z=13/3$. It is also obvious that any further branching of the problem will not improve the value of the objective function, as the next subdivision will impose the restrictions $X_1\le 3$ and $X_1\ge 4$ respectively. Then the Optimal Solutions are $X_1=3$ and $X_2=1$and $X_1=4$,$X_2=0$ respectively. Both of these solutions give the max value of Z=4. It may be noted that there does not exist any feasible solution to sub problem 5.

Hence, over all the maximum values of the objective function the maximum is Z=4 and the integer valued solution is any of the three $X_1=2$ and $X_2=2$ or $X_1=3$ and $X_2=1$ or $X_1=4$, $X_2=0$.

Results of Computational Testing

Results from one of the test models-the so-called 'fixed changes' MILP model are especially interesting. Consider the following IP model

Maximize $Z = 7X_1+9X_2$

Subject to $-X_1+3X_2\le 6$, $7X_1+X_2\le 35$, $X_1 \ge 0$, $X_2\le 7$, X_1, X_2 are integers.

The above problem can be rewritten as follows

Minimum $Z = -7X_1-9X_2$

Subject to $-X_1+3X_2+X_3=6$, $7X_1+X_2+X_4 =35$, $X_1 \ge 0$, $X_2\le 7$, X_1, $X_2$are integers.

At the starting iteration we can very well consider $Z^* =0$ to be the lower bound for Z, since all $X_j=0$ are feasible. The master list contains only the LPP. The given problem is designated as problem 1. Initially, the V list is empty, $Z^*\to\infty$

Step 1: Solve the underlying LP model using the Revised Simplex method, determine the Optimal Solution Z=63, $X_1=9/2$, $X_2=7/2$

$$X_D^{(1)} = d^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Since the solution is not integer valued, select X_1, then since $[X_1^*] = 9/2 = 4$ place on the master list the following two additional problems

Sub model 2

Maximize $Z = 7X_1+9X_2$

Subject to $-X_1+3X_2\le 6$, $7X_1+X_2\le 35$, $5\le X_1\le 7$, $0\le X_2\le 7$ X_1, $X_2$are integers.

Sub model 3

Maximize $Z = 7X_1+9X_2$

Subject to $-X_1+3X_2\le 6$,$7X_1+X_2\le 35$,$0\le X_1\le 4$, $0\le X_2\le 7$ X_1, $X_2$are integers.

Step 2: Push the twin models, each with opposite integer bounds on to the V-list

$V = \{5\le X_1\le 7, 0\le X_1\le 4\}$

Step 3: Select the first model in V. The model to be solved as

	L	U	
X_1	5	7	
X_2	0	∞	Sub model (2)
X_3	0	∞	
X_4	0	∞	

And $V = \{5\le X_1\le 7\}$ gives Z = 35, $X_1=5$, $X_2=0$ (Solution to sub model 2)

And $V = \{0\le X_1\le 4\}$

	L	U
X_1	0	4
X_2	0	∞
X_3	0	∞
X_4	0	∞

Sub model (3)

Step 1: Start from the current solution to solve the twin model. We find submodel 3 is available and determines the following optimum feasible solution to it Z= 58, X_1=4, X_2=10/3 Since the solution is not integer valued, select X_2. Then X_2^*=[10/3] = 3, we additional problems on the master list are placed

<u>Sub model 4</u>

Maximize $Z = 7X_1+9X_2$

Subject to $-X_1+3X_2 \leq 6$, $7X_1+X_2 \leq 35$, $0 \leq X_1 \leq 4$, $4 \leq X_2 \leq 7$ X_1, X_2 are integers.

<u>Sub model 5</u>

Maximize $Z = 7X_1+9X_2$

Subject to $-X_1+3X_2 \leq 6$, $7X_1+X_2 \leq 35$, $0 \leq X_1 \leq 4$, $0 \leq X_2 \leq 3$ X_1, X_2 are integers.

Step 2: Push two twin models each with opposite integer bounds onto the V list

$V = \{0 \leq X_2 \leq 3, 4 \leq X_2 \leq 7\}$

Step 3: Select the first model in V. The model to be solved as:

	L	U
X_1	0	4
X_2	0	3
X_3	0	∞
X_4	0	∞

and $V=\{4 \leq X_2 \leq 7\}$. $Z^* = -\infty$ At this step L=0=d so the current solution (not optimum) to this model.

Step 1: Start from the current solution to solve the twin model. We determine the following Optimal Solution to this problem Z = 55, X_1=4, X_2=3.

At this point the process discovers that an integer solution is found, and proceeds go to step 3.

Step 3: Pop the next bound $X_1 \leq 4$ from the V list. This sub model has no feasible solution. Hence go to Step 4.

Step 4: The incumbent solution Z = 55 is optimal for the IP model

The above integer solution obtained is the optimum solution of the given problem. It reduces the round off error. Results of the problem give the integer solution.

The theoretically best decisive rule for making any such selection should embody the following characteristics

- It should be computationally simple
- It should tend to minimize the number of iterations required to obtain an optimal solution
- It should be computationally efficient on the basis of time

The above proposed IEA possess the above characteristics

Criteria for Adding New Constraint

In IEA method, adding new constraint is a different one from the other integer programming methods. Initially solve the given Integer problem by linear programming method i.e. Revised simplex method. We get the solution for variables like $X_j : j \in I$. some of the variables may have fractional or non-integer solution. Choose, from among those variables $X_j : j \in I$. that does not have integral values at the node, one variable to be used to form the branching constraints. An easily implemented rule for this choice is to use the variable whose value has the largest fractional part. We can write

$X_{Bi} = [X_{bi}] + f_i$.

Where f_i lies between 0 and 1 (ie) $0<f_i<1$. Since X_j must have an integral value, it must satisfy either $X_j \leq [X_{Bi}]$ or $X_j \geq [X_{Bi}] +1$

We create two new mixed integer (formation of new nodes) problems represented by the node under consideration in the provisional steps. One problem is formed by ADDING CONSTRAINT $X_j \leq [X_{Bi}]$ and the other problem is formed by ADDING CONSTRAINT $X_j \geq [X_{Bi}] +1$. Solve each of these problems as a linear programming problem using the dual simplex or Revised simplex method.

<u>Example</u>

Maximize $Z=7X_1+3X_2$

Subject to $2X_1+5X_2 \leq 30$, $8X_1+3X_2 \leq 48$, $X_1, X_2 \geq 0$

We solve the related linear programming problem and obtain the final tableau. From this we see that an optimal solution is $X_1=4\frac{7}{17}$, $X_2 = 4\frac{4}{17}$ $Z = 43\frac{10}{17}$

We choose X_1 as the branching variable, since it has the largest fractional part. The constraints to be added are $X_1 \leq 4$, $X_2 \geq 5$.

	X_1	X_2	X_3	X_4	
X_2 X_1	0 1	1 0	4/17 -3/34	-1/17 5/34	72/17 75/17
Z	0	0	3/34	29/34	741/17

We add these constraints in turn to the final tableau

$$X_1 = \frac{75}{17} + \frac{3}{34}X_3 - \frac{5}{34}X_4$$

$\Rightarrow X_1 + U_1 = 4$

$$\frac{3}{34}X_3 - \frac{5}{34}X_4 + U_1 = 4 - \frac{75}{17} = \frac{-7}{17}$$

$$\frac{3}{34}X_3 - \frac{5}{34}X_4 + U_1 = \frac{-7}{17} \qquad \rightarrow A$$

and for $X_1 \geq 5$

$$\frac{3}{34}X_3 - \frac{5}{34}X_4 + U_1 = 5 + \frac{75}{17} = \frac{-10}{17}$$

$$\frac{3}{34}X_3 - \frac{5}{34}X_4 + U_1 = \frac{-10}{17} \qquad \rightarrow B$$

Where A and B are two new constraints added to the original problem.

Basis for Comparison

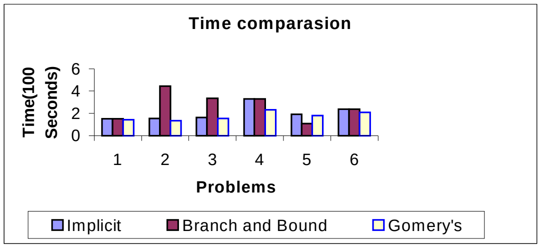
Comparisons based on number of iterations and time for a few typical problems. It may be observed that the computational savings obtained on the basis of the number of iterations is invariably greater than that obtained based on time. And the main aspect of comparison is optimal solution, which is compared with other two algorithms or programs. For any given problem the implicit branch and bound has always been superior to the other integer programming methods on the basis of the number of multiply/divide operations and processing time.

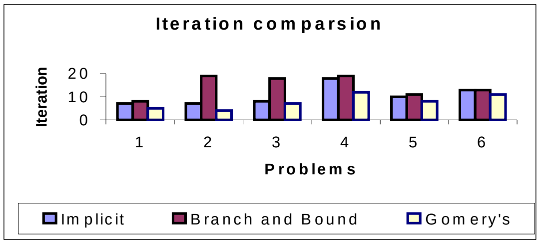
Table 1.Time Comparison

Problem	Branch and Bound	Gomery's method	Implicit method
Problem 1	1.53	1.53	1.43
Problem 2	1.54	4.45	1.34
Problem 3	1.65	3.35	1.56
Problem 4	3.29	3.3	2.34
Problem 5	1.92	1.100	1.8
Problem 6	2.37	2.36	2.1

Table 2. Iteration Comparison

Problem	Branch and Bound	Gomery's method	Implicit method
Problem 1	7	8	5
Problem 2	7	19	4
Problem 3	8	18	7
Problem 4	18	19	12
Problem 5	10	11	8
Problem 6	13	13	11





Conclusions

This paper deals with the round-off error and computer memory requirements of Branch and bound algorithms. IEA increases the efficiency (Speed) of the branch and bound algorithm and avoids the round off errors. The accuracy of the IA on small test models especially designed to be difficult to solve, and whose optimal solutions are known is good. Running on an 80286 based microcomputer. The algorithm found the correct solution in all cases. The fact that it required many branching to solve this small model is strong evidence of its invulnerability to round-off error. And the fact that it produced an optimal solution within the 640 KB of memory of the microcomputer (actually, it only required 100 KB) demonstrates its very modest computer memory requirements.

Reported computational experiences indicate that the use of the surrogate constraint is effective in improving the computation time. However because implicit enumeration investigates all 2^n binary points, the solution time varies almost exponentially with the number of variables n.

References

1. A.Land and A.Doig, An automatic method for solving discrete programming problems, Econometrica 28, 497-520
2. R.Dakin, A tree-search algorithm for mixed integer-programming problems, comput. J.*, 250-255
3. YOULING LIN, LARY M AUSTIN AND JAMES R BURNS, AnImplicit *Branch and Bound Algorithm for Mixed Integer Linear Programming,* Computers Opns Res Vol 17 No 5, Page 457-464, 1990.
4. L.M.Austin and B.Ruparel, The mixed cutting plane algorithm for all-integer programming. Computers opns Res 13,395-401 (1986)
5. Dr. B.S. GOEL and Dr. S.K. MITTAL , *OR*, Pragati Prakashan ,Meerut, 1995.
6. P.K.GUPTA and D.S.HIRA *Operation Research,* S.Chand & Co Ltd., NewDelhi. 1994.
7. H.A Taha, Operations Research, An Introdution 3rd edn, pp 250-255. Macmillan, New york(1982)
8. F.S. Hillier and G.J>Lieberman, Introduction to Operations Research, 4th Edn, pp274-276, Holden-Day, Oakland,CA(1986)
9. P.Ghandforoush, A constructive primal-dual cutting plane algorithm for integer programming. Unpublished P.hd disertation, College of Administration, Texas Tech University
10. C.A Trauth and R.E.D. Woolsey, Integer linear programming: a study in computational efficiency, Mgmt Sci 15, 481-493.

Appendix

Numerical Examples

1.Maximize $Z = 3X_1 + 2X_2 + 5X_3$
Subject to X_1+2X_2+X3 ³ 430
$3X_1+2X_3$£ 460, X_1+4X_2£ 420, X_1,X_2,X_3 ³ 0
Solution: $Z=1350, X_1=0, X_2=100, X_3=230$

2.Maximize $Z=7X_1+3X_2$
Subject to $2X_1+5X_2$£30,
$8X_1+3X_2$£48, X_1,X_2 ³ 0
Solution : $Z=42, X_1=6, X_2=0$

3.Maximize $Z = 7X_1 + 9X_2$
Subject to $-X_1+3X_2$£ 6
$7X_1+X_2$£ 35, X_1,X_2 ³ 0
Solution: $Z=55, X_1=4, X_2=3$

4. Maximize $Z = X_1 + 4X_2$
Subject to $2X_1+4X_2$£ 7
$5X_1+3X_2$ ³ 15, X_1,X_2 ³ 0
Solution: $Z=5, X_1=1, X_2=1$

5.Maximize $Z = 6X_1 + X_2$
Subject to $13X_1+2X_2$£ 25
$8X_1+3X_2$£ 34, X_1+4X_2£ 22,X_1,X_2 ³ 0
Solution: $Z=17, X_1=2, X_2=5, X_3=9$

6.Maximize $Z = 3X_1 + 2X_2$
Subject to $X_1+X_2-X_3$£1, X_1+2X_2£10,
X_1+X_2£ 7, X_1,X_2,X_3 ³ 0
Solution: $Z=18, X_1=4, X_2= 3, X_3=6$

The tree for problems constructed by the branch and bound algorithm i.e IEA for the example.