An Advance Technique for Solving Assignment Problems

P.R. Sharma[*], S. Shukla[**]
[*]Department of Mathematics, University of Rajasthan, Jaipur (India)
[**]S. S. Jain Subodh P.G. College, Jaipur (India)
Email Id.: shailjpr@yahoo.com , shailjpr41184@gmail.com

Abstract: The assignment problem is a typical combinatorial optimization problem. Classical Assignment Problem (AP) is a well-known topic world-wide. In this problem c_{ij} denotes the cost for assigning the j^{th} job to the i^{th} person. This cost is usually deterministic in nature. But in realistic situations, it may not be practicable to know the precise values of these costs. We consider many classical problems from location theory which may serve as theoretical models for several logistic problems such that some linear or quadratic function attains its minimum. It turns out that linear objective function yields a linear assignment problem, which can be solved easily by several primal-dual methods like Hungarian method, Shortest augmenting path method etc.
Aim of the paper is to investigate a new approach to solve assignment problems of different types. This method proposes momentous advantages over similar methods.

Keywords: Assignment Problem, Branch, Bound Technique.

Introduction:

The assignment problem is one of the earliest applications of linear integer programming problem. A diversity of practical problems turn out to be a special illustration of the assigning problem, i.e. a problem, where one looks for an assignment of members of set A to members of set B such that some function attains its optimum. We may assume that the numbers of elements in sets A and B are equal and that we want to assign exactly one element from A to each element from B. Solving such problem often means that we have to evaluate some function for each assignment. Since there are n! Possible assignments, where n is the number of elements in A, this may lead to a very hard problem. Different methods have been presented to solve assignment problem by Goel and Mittal (1982), Bazarra et al. (2005) and Hamdy (2007).

A considerable number of methods have been so far presented for assignment problem in which, the best known, most widely used, and most written about method for solving the assignment problem is the "Hungarian Method", originally suggested by Kuhn in 1955. Ford and Fulkerson (1957) provided vital ideas for the untimely approaches used in solving network flow problems, extended to solve the transportation problem [Munkres (1957)] and generalized to solve the linear programming problem [Dantzig et al. (1956)]. It is a dual method with a feasible assignment being obtained only at the last computational step.

Aim of the paper is to apply Branch and Bound technique for all type of assignment problems. The proposed method is a methodical process, easy to apply and can be exploited for all types of assignment problems with maximize or minimize objective functions.

Mathematical Formulation of Assignment Problem:

Let there be n persons and n jobs. Each job must be done by exactly one person and one person can do, at most, one job. If the cost of doing j^{th} job by i^{th} person is c_{ij}, then the cost matrix is given by the following table:

jobs Person	1	2	3	………. J	………..n
1	c_{11}	c_{12}	c_{13}	………. c_{1j}	……….. c_{1n}
2	c_{21}	c_{22}	c_{23}	………. c_{2j}	……….. c_{2n}
3	c_{31}	c_{32}	c_{33}	………. c_{3j}	……….. c_{3n}
……….. ……….. i	…… …… c_{i1}	….. ….. c_{i2}	….. ….. c_{i3}	………… ………… ……….. c_{ij}	………… ………… ………… c_{in}
……….. ……….. n	….. ….. c_{n1}	….. ….. c_{n2}	…. …. c_{n3}	……….. ……….. ……….. c_{nj}	…………. …………. …………. c_{nn}

Table 1

The problem is to assign the persons to the jobs so that the total cost of completing all jobs become minimum.

We introduce x_{ij}, where

$$x_{ij} = \begin{cases} 1, \text{ if the person i is assigned the job j; i,j = 1, 2, ..., n} \\ 0, \text{ otherwise} \end{cases} \qquad \text{………. (1)}$$

Corresponding to the $(i\ j)^{th}$ event of assigning person i to job j, the constraint

$\sum_{i=1}^{n} x_{ij} = 1,\ j = 1, 2, ..., n$ i.e. each job must be done by exactly one person, and the constraint $\sum_{j=1}^{n} x_{ij} = 1, \text{i} = 1, 2, ..., n$ means each person must be assigned at most one job.

Thus the model for crisp Assignment Problem (AP) is given by

Model 1:

$$Min\ z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \qquad \text{………. (2)}$$

$$\text{subject to}\begin{cases}\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,....,n \\ \sum_{j=1}^{n} x_{ij} = 1, \quad \text{i} = 1,2,....,n \\ x_{ij} = 0 \text{ or } 1, \ i,j = 1,2,...,n \end{cases} \qquad .... (3)$$

This cost c_{ij} is usually deterministic in nature. But in real situations, it may not be practicable to know the precise values of these costs. In such an uncertain situation, instead of exact values of costs, if the preferences for assigning the j^{th} job to the i^{th} person is known in the form of composite relative degree (d_{ij}) of similarity to ideal solution (maximum degree indicates most preferable combination), then c_{ij} can be replaced by d_{ij} in the classical assignment problem in the maximization form which can be solved by any standard procedure (e.g. Hungarian method or by any software) to get the optimal assignment. In that case the model for the preference AP becomes

Model 2:

$$Max\ z = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} x_{ij} \qquad .... (4)$$

$$\text{subject to}\begin{cases}\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,....,n \\ \sum_{j=1}^{n} x_{ij} = 1, \quad \text{i} = 1,2,....,n \\ x_{ij} = 0 \text{ or } 1, \ i,j = 1,2,...,n \end{cases} \qquad .... (5)$$

Method of Solution:

In this paper, the assignment problem is solved by Branch and Bound Algorithm. Using curtailed enumeration technique. The terminologies of the branch and bound technique applied to the assignment problems are presented below:

Let k be the level number in the branching tree (for root node, it is 0), σ be an assignment made is the current node of a branching tree. p_{σ}^{k} be an assignment at level k of the branching tree, A be the set of assigned cells (fractional assignment) up to the node p_{σ}^{k} from the root node (set of i and j values with respect to the assignment cells up to the node p_σ^k from the root node), and v_σ be the lower bound of the fractional assignment A up to p_σ^k , such that

$$v_\sigma = \sum_{i,j\in A} c_{ij} + \sum_{i\in X}\left(\sum_{j\in Y} Min\ c_{ij}\right) \quad \text{……… (6)}$$

where c_{ij} is the cell entry of the cost matrix with respect to the i^{th} row and j^{th} column, X is the set of rows which are not removed up to the node p_σ^k from the root node in the branching tree, and Y is the set of column which are not removed up to the node p_σ^k from the root node in the branching tree.

Branching guidelines

1. At level k, the row marked as k of the assignment problem will be assigned with the best column of the assignment problem.
2. If there is a tie on the lower bound, then the terminal node at the lower most level is to be considered for further branching.
3. Stopping rule: - If the minimum lower bound happens to be at any one of the terminal nodes at the $(n-1)^{th}$ level, the optimality is reduced, and then the assignment on the path from the root node to that node along with the missing pair of row column arrangement will from the optimum solution.

Types of Assignment Problems:

There are three types of Assignment Problems.

Type-I: In maximization problems jobs effectiveness is frequently measured by profit instead of cost. When a worker is assigned to different jobs, the profit contribution often differs from jobs to jobs. This difference arises from workers capability and experience on a particular jobs, as well as the different nature of jobs to be assigned. Therefore job effectiveness of the worker is expressed in terms of profit matrix.

Except for one transformation an assignment problem in which the objective is maximize total payoff measures can be solved by the Branch and Bound Algorithm.

The transformation involves subtracting all the entries of the original payoff table from the maximum entry of that table. The transformed entries give us the relative costs and the problem then becomes a minimization problem. Once the optimal assignment for the transformed problem is obtained. The total measure of the original payoff matrix can be found by those cells to which the assignment has been made. In these types of problem the number of persons to be assigned and number of jobs were assumed to be the same. Such as assignment problem is known as balanced assignment problem.

Type-II: If the number of persons is different from number of jobs, the assignment problem is said to be unbalanced problem. If the number of jobs is less than the number of persons, some of the persons cannot be assignment any job. In such problems one or more dummy jobs of zero duration are introduced to make the assignment problem balanced. On the other hand if the number of persons is less than the number of jobs than we add one or more dummy persons with duration time zero to balance the assignment problem. The balanced problem then can be solved by using the Branch and Bound Techniques.

Type-III: In some cases, a certain worker cannot be assigned a particular job. The reasons for impossible assignments are numerous. Back of required skills, deficiency in technical know-how, improper training and physical inability are only a few many reasons. For solving such type of assignment problems, infinite cost is put in the cell where no assignment is possible. The remaining procedure is exactly the same as the ordinary assignment problems.

Numerical Examples:

Example–1. The jobs 1, 2, 3 are to be assigned three machines 1, 2,3 the processing cost (Rs/-) are as given in the matrix shown below, find the allocation which will minimize the overall processing cost

		Machine		
		1	2	3
Jobs	1	19	28	31
	2	11	17	16
	3	12	15	13

Solution: Initially, no job is assigned to any machine so the assignment(σ) at the root node(level 0)of the following branching tree is a null set and the corresponding lower bound v_σ is also 0, as shown is Figure 1:

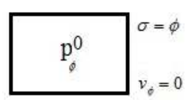

Figure1. Branching tree at the root node

Further branching: The three different sub-problems under the root node are shown in the Figure 2, the lower bound for each of the sub-problems is shown by the right side of it.

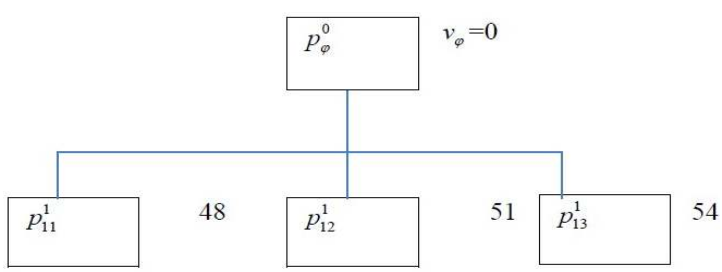

Figure 2. Branching tree after from p^0_φ

Calculation for lower bound:

Lower bound for p^1_{11}

$$v_\sigma = \sum_{i,j\in A} c_{ij} + \sum_{i\in X} \left(\sum_{j\in y} \min c_{ij}\right). \qquad \text{………. (7)}$$

Where σ = {(11)}, A = {(11)}, X= (2, 3), Y= (2, 3).

Then

$$v_{11} = c_{11} + \sum_{i\in(2,3)} \left(\sum_{j\in(2,3)} \min c_{ij} \right) \qquad \text{……….. (8)}$$

$= c_{11} + c_{23} + c_{33}$ = 19+16+13=48.

Lower bound for p_{12}^{1}

σ = {(12)}, A = {(12)}, X= (2, 3), Y= (1, 3).

Then

$$v_{12} = c_{12} + \sum_{i\in(2,3)} \left(\sum_{j\in(1,3)} \min c_{ij} \right) \qquad \text{……….. (9)}$$

$= c_{12} + c_{21} + c_{31}$ = 28+11+12 = 51.

Lower bound for p_{13}^{1}

σ = {(13)}, A = {(13)}, X= (2, 3), Y= (1, 2).

Then

$$v_{13} = c_{13} + \sum_{i\in(2,3)} \left(\sum_{j\in(1,2)} \min c_{ij} \right) \qquad \text{……….. (10)}$$

$= c_{13} + c_{21} + c_{31}$ = 31+11+12 = 54.

Further branching: Further branching is done from the terminal node which has the least lower bound. At this stage, the nodes p_{11}^{1}, p_{12}^{1}, and p_{13}^{1} are the terminal nodes. Among these nodes, the node p_{11}^{1} has the least lower bound. Hence, further branching from this node is shown in Figure 3. The lower bound of each of the newly created node is shown by the right side of it.

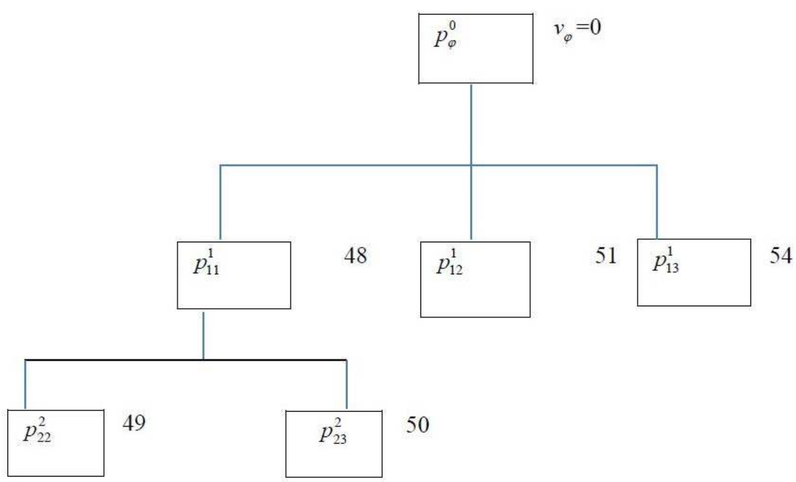

Figure 3. Branching tree after from p_{11}^{1}

Calculation for lower bound:

Lower bound for p_{22}^2

σ = {(22)}, A = {(11), (22)}, X= (3), Y= (3).

Then

$$v_{22} = c_{11} + c_{22} + \sum_{i \in 3} \left(\sum_{j \in 3} \min c_{ij} \right) \qquad \text{……….} (11)$$

$= c_{11} + c_{22} + c_{33}$ = 19+17+13 = 49.

Lower bound for p_{23}^2

σ = {(23)}, A = {(11), (23)}, X= (3), Y= (2).

Then

$$v_{23} = c_{11} + c_{23} + \sum_{i \in 3} \left(\sum_{j \in 2} \min c_{ij} \right) \qquad \text{……….} (12)$$

$v_{23} = c_{11} + c_{23} + c_{32}$ =19+16+15=50.

Further branching: At this stage, the nodes p_{12}^1, p_{13}^1, p_{23}^2 and p_{22}^2 are the terminal nodes. Among these nodes, there is one node with the least lower bound of 49. So the node p_{22}^2 which is at the bottom-most level is considered for further branching. Since this node lies at $(n-1)^{th}$ (level k=2) of the branching tree, where n is the size of the assignment problem, optimality is reached. The corresponding solution is traced from the root node to the node p_{22}^2 along with the missing pair of job and operator combination (3, 3), so optimal solution is presented through the following table.

Job	Machine	Cost
1	1	19
2	2	17
3	3	13

Hence total cost = 19+17+13 = 49 Rs.

Example-2. A methods engineer wants to assign four new methods to three works centers. The assignment of the new methods will increase production and they are given below. If only one method can be assigned to a work center, determine the optimum assignment:

Increase in (production) unit

Methods

	A	B	C
1	10	7	8
2	8	9	7
3	7	12	6
4	10	10	8

Solution: First we will convert maximum matrix into minimum matrix, for this subtracting all the elements of given matrix from maximum element (12), then we have minimum matrix as following:

	A	B	C
1	2	5	4
2	4	3	5
3	5	0	6
4	2	2	4

Now we will introduce a dummy column for making a square matrix, which is balanced assignment problem is given below:

	A	B	C	D
1	2	5	4	0
2	4	3	5	0
3	5	0	6	0
4	2	2	4	0

Initially no method is assigned to any production so the assignment (σ) at the root node of the following branching tree is a null set and the corresponding lower bound v_σ is also 0, as shown in Figure 4.

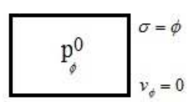

Figure 4. Branching tree at the root node

Further branching: The four different sub-problems under the root node are shown in the Figure 5, the lower bound for each of the sub-problems is by the right side of it.

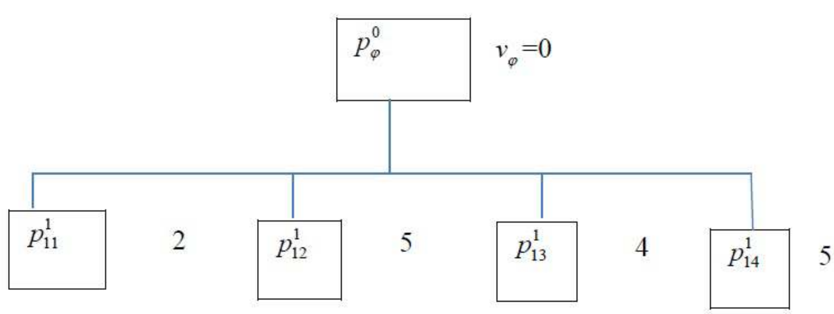

Figure 5. Branching tree at the root node p^0_ϕ

Calculation for lower bound:

Lower bound for p_{11}^{1}

$$v_\sigma = \sum_{i,j\in A} c_{ij} + \sum_{i\in X} (\sum_{j\in y} \min c_{ij}). \qquad \text{………. (13)}$$

Where σ = {(11)}, A = {(11)}, X= (2, 3, 4), Y= (2, 3, 4).

$$v_{11} = c_{11} + \sum_{i\in(2,3,4)} \left(\sum_{j\in(2,3,4)} Min\ c_{ij} \right) \qquad \text{………. (14)}$$

$= c_{11} + c_{24} + c_{34} + c_{44}$ =2+0+0+0=2.

Lower bound for p_{12}^{1}

σ = {(12)}, A = {(12)}, X= (2, 3, 4), Y= (1, 3, 4).

Then

$v_{12} = c_{12} + c_{24} + c_{34} + c_{44}$ ==5+0+0+0=5. ……….. (15)

Lower bound for p_{13}^{1}

σ = {(13)}, A = {(13)}, X= (2, 3, 4), Y= (1, 2, 4).

Then

$v_{13} = c_{13} + c_{24} + c_{34} + c_{44}$ =4+0+0+0=4. ………. (16)

Lower bound for p_{14}^{1}

σ = {(14)}, A = {(14)}, X= (2, 3, 4), Y= (1, 2, 3).

Then

$v_{14} = c_{14} + c_{22} + c_{32} + c_{42}$ =0+3+0+2=5. ……….. (17)

Further branching:-Further branching is done from the terminal node which has the least lower bound. At this stage, the nodes p_{11}^{1}, p_{12}^{1}, p_{13}^{1} and p_{14}^{1} are the terminal nodes. Among these nodes, the node p_{11}^{1} has the least lower bound of 2. Hence, further branching from this node is shown as Figure 6. The lower bound of each of the newly created node is shown by the right side of it.

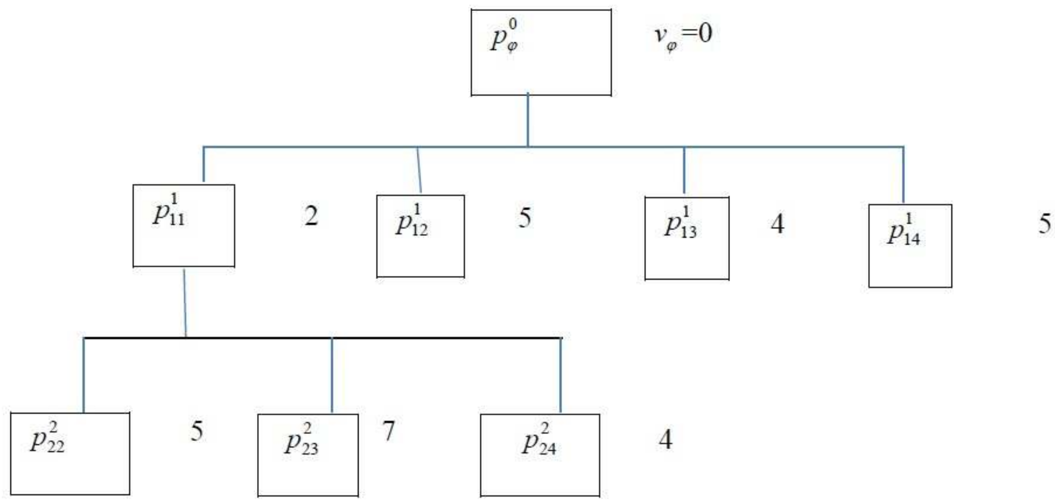

Figure 6. Branching tree after branching from p_{11}^{1}

Calculation for lower bound:

Lower bound for p_{22}^{2}

σ = {(22)}, A = {(11), (22)}, X= (3, 4), Y= (3, 4).

Then

$v_{22} = c_{11} + c_{22} + c_{34} + c_{44}$ =2+3+0+0=5. ……….. (18)

Lower bound for p_{23}^{2}

σ = {(23)}, A = {(11), (23)}, X= (3, 4), Y= (2, 4).

Then

$v_{23} = c_{11} + c_{23} + c_{34} + c_{44}$ =2+5+0+0=7. ………… (19)

Lower bound for p_{24}^{2}

σ = {(24)}, A = {(11), (24)}, X= (3, 4), Y= (2, 3).

Then

$v_{24} = c_{11} + c_{24} + c_{32} + c_{42}$ =2+0+0+2=4. ……….. (20)

Further branching: Further branching is done from the terminal node which has the least lower bound. At this stage, the nodes p_{12}^{1} **,** p_{13}^{1}, p_{14}^{1} , p_{22}^{2} , p_{23}^{2} and p_{24}^{2} are the terminal nodes. Among these nodes, two nodes have the least lower bound of 4. So the node p_{24}^{2} which is at the bottom most level is considered for further branching from this node is shown as Figure 7. The lower bound of newly created node by the right side of it.

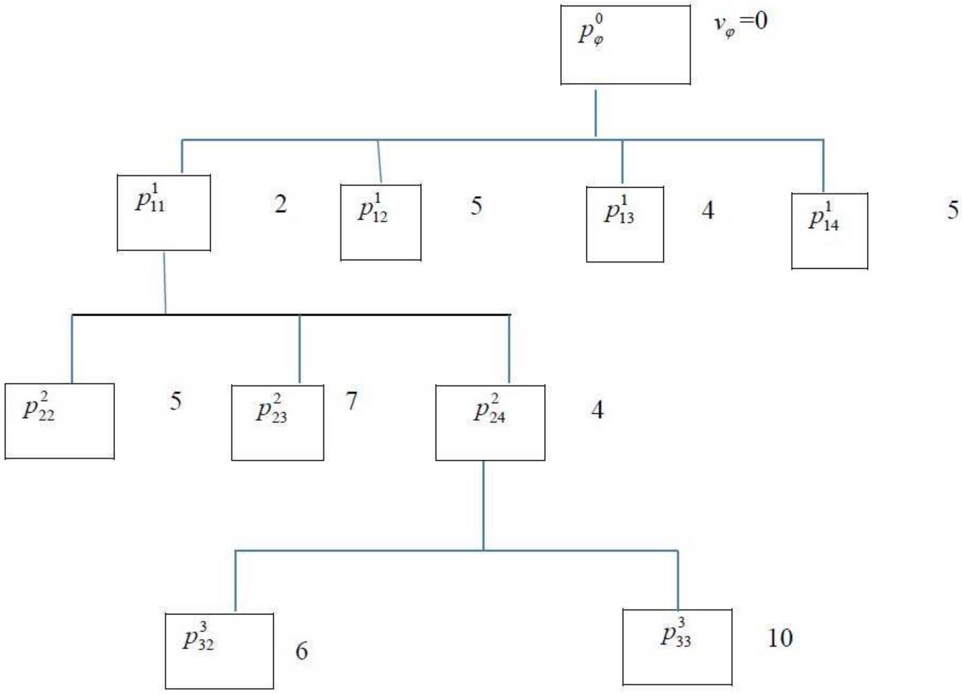

Figure 7. Branching tree after branching from p_{24}^2

Calculation for lower bound:

Lower bound for p_{32}^3

σ = {(32)}, A = {(11), (24), (32)}, X= (4), Y= (3).

Then

$v_{32} = c_{11} + c_{24} + c_{32} + c_{43}$ =2+0+0+4=6. ……….. (21)

Lower bound for p_{33}^3

σ = {(33)}, A = {(11), (24), (33)}, X= (4), Y= (2).

Then,

$v_{33} = c_{11} + c_{24} + c_{33} + c_{42}$ =2+0+6+2=10 .... (22)

Further branching: At this stage, the nodes p_{12}^1, p_{13}^1, p_{14}^1, p_{23}^2, p_{22}^2, p_{32}^3 and p_{33}^3 are the terminal nodes. Among these nodes, there is one node p_{13}^1 with the least lower bound but not bottom-most node and p_{32}^3 is bottom-most node. Hence, further branching from p_{32}^3 node. Since this node lies at $(\mathrm{n}-1)^{th}$ level of the branching tree, where n is the size of the assignment problem, optimality is reached. Hence, optimal solution is

Method	Production	Unit
1	A	10
2	B	0
3	C	12
4	D	8

Hence total units =10+0+12+8=30.

Example-3. Secretary of a school is talking bids on the city's four school bus routes. Four companies have made the bids as detailed in the following table.

Company	Bids			
	A	B	C	D
C_1	7000	8000	---	---
C_2	---	7000	---	7000
C_3	6000	---	5000	---
C_4	---	---	7000	8000

Suppose each bidder can be assigned only to one route. Use the assignment model to minimize the school's cost of running the four bus routes.

Solution: First we are taking highest cost ( ) for unassigned routes. The problem can be represented as

	A	B	C	D
C_1	7000	8000		
C_2		7000		7000
C_3	6000		5000	
C_4			7000	8000

$\Rightarrow 1000 \times$

7	8		
	7		7
6		5	
		7	8

Initially no company is assigned to any bidder, so the assignment (σ) at the root node (level 0) of the following branching tree is a null set and the corresponding lower bound v_σ is also 0, as shown in Figure 8.

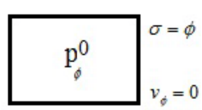

Figure 8. Branching tree at the root node

Further Branching: The four different sub-problems under the root node are shown in Figure 9. The lower bound for each of the sub-problems is shown by the right side of it.

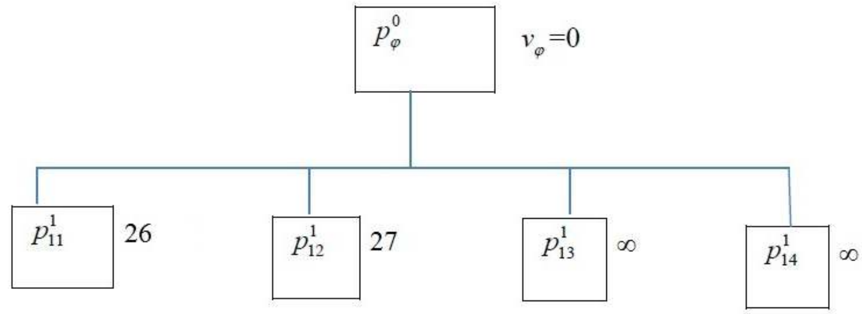

Figure 9. Branching tree after branching from p_ϕ^0

Calculation for lower bound:

Lower bound for p_{11}^1

$$\sigma = \{(11)\}; A = \{(11)\}; x = (2,3,4), y = (2,3,4).$$
$$v_{11} = c_{11} + c_{22} + c_{33} + c_{43} = 7+7+5+7 = 26. \quad \text{.......... (23)}$$

Lower bound for p_{12}^1

$$\sigma = \{(12)\}; A = \{(12)\}; x = (2,3,4), y = (1,3,4).$$
$$v_{12} = c_{12} + c_{24} + c_{33} + c_{43} = 8+7+5+7 = 27. \quad \text{.......... (24)}$$

Further Branching: At this stage, the nodes $p_{11}^1, p_{12}^1, p_{13}^1$ and p_{14}^1 are terminal node. Among these nodes, the node p_{11}^1 has the least lower bound. Hence, further branching from this node is shown in Figure 10. The lower bound of newly created nodes by the right side of it.

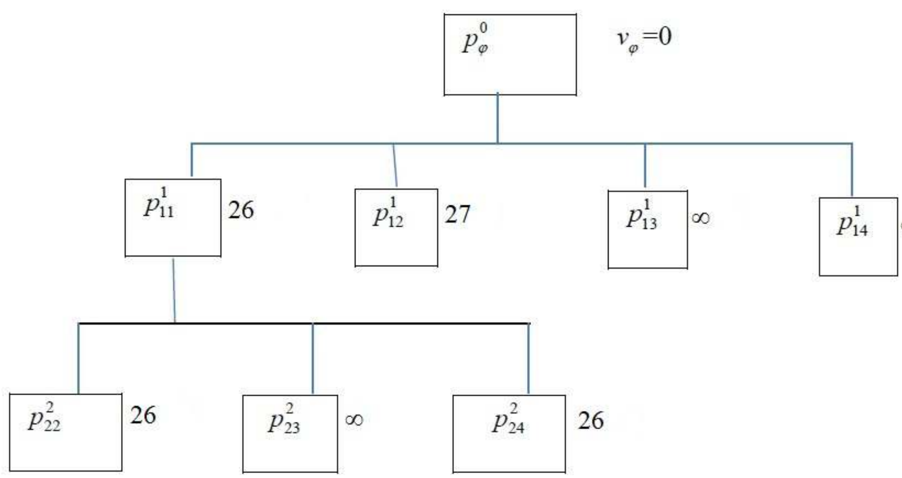

Figure 10. Branching tree after branching from p_{11}^1

Calculation for lower bound:

Lower bound for p_{22}^{2}

$$\sigma = \{(22)\}; A = \{(11),(22)\}; x = (3,4), y = (3,4).$$
$$v_{22} = c_{11} + c_{22} + c_{33} + c_{43} = 7 + 7 + 5 + 7 = 26. \qquad \text{.......... (25)}$$

Lower bound for p_{24}^{2}

$$\sigma = \{(24)\}; A = \{(11),(24)\}; x = (3,4), y = (2,3).$$
$$v_{24} = c_{11} + c_{24} + c_{33} + c_{43} = 7 + 7 + 5 + 7 = 26. \qquad \text{.......... (26)}$$

Further Branching: At this stage, the nodes $p_{12}^{1}, p_{13}^{1}, p_{14}^{1}, p_{22}^{2}, p_{23}^{2}$ and p_{24}^{2} are terminal node. Among these nodes, two nodes have least lower bound of 26. The node p_{22}^{2} is the first least lower bound so that further branching considering from this node shown in Figure 11. The lower bound of newly created nodes by the right side of it.

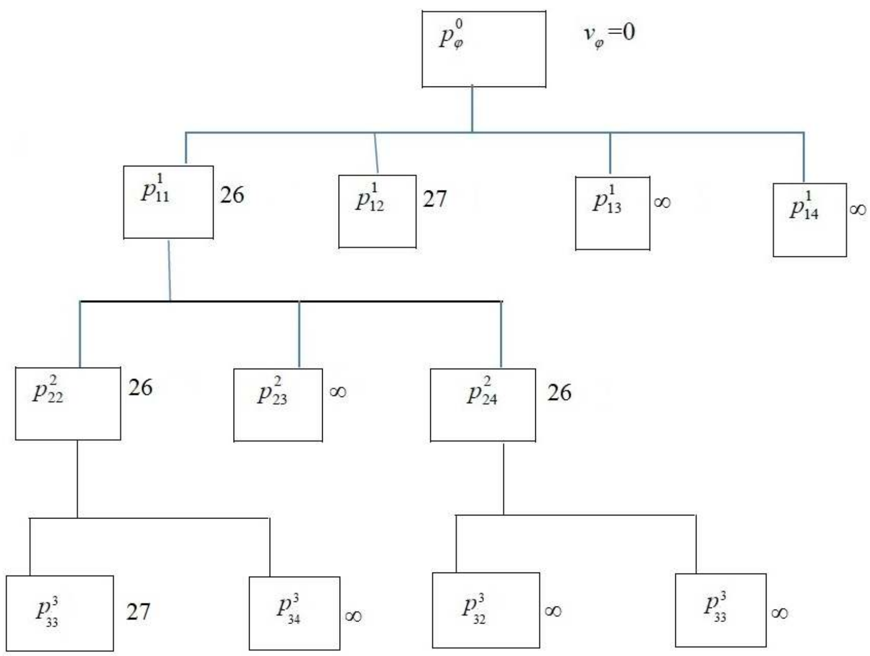

Figure 11. Branching tree after branching from p_{22}^{2}

Calculation for lower bound:

Lower bound for p_{33}^{3}

$$\sigma = \{(33)\}; A = \{(11), (22), (33)\}; x = (4), y = (4).$$
$$v_{33} = c_{11} + c_{22} + c_{33} + c_{44} = 7 + 7 + 5 + 8 = 27. \quad \text{.......... (27)}$$

Further Branching: At this stage, the nodes p_{12}^{1}, p_{13}^{1},p_{14}^{1}, p_{23}^{2}, p_{24}^{2}, p_{33}^{3} and p_{34}^{3} are terminal node. Among these nodes, the node p_{24}^{2} has least lower bound but this is not bottom most and after branching from this node we have nodes p_{32}^{3} and p_{33}^{3}. These nodes have highest lower bound so that this not for further branching. Hence p_{33}^{3} is second least lower bound and bottom most node at level three. Since this node lies at $(n-1)^{th}$ level of the branching tree, where n is the size of the assignment problem, optimality is reached.

Hence optimal solution is

Company	route	Cost
C1	A	7000
C2	B	7000
C3	C	5000
C4	D	8000

Hence total cost= (7+7+5+8) ×1000=27000 Rs.

Conclusion

In this paper, Branch and Bound technique is used to solve assignment problem. This method is applicable for all kind of assignment problems, whether maximize or minimize objective function. This technique is easy to apply and consume less time comparative to another techniques.

Acknowledgement: The authors are thankful to Dr. Sandeep gupta for his valuable help in preparation of the paper.

References

Goel B.S., Mittal, S. K. (1982): Operations Research. (Pragati Prakashan). 2405-2416.

Dantzig G. B., Ford, L. R., Jr. and Fulkerson D. R., (1956): A Primal-Dual Algorithm for Linear Programs, in Linear Inequalities and Related Systems. *Annals of Mathematics Study No. 38*, (Eds) Kuhn and Tucker, (Princeton University Press, Princeton).

Ford L. R., Jr. and Fulkerson D. R. (1957): A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem. *Canadian Journal of Mathematics*. **9**, 210-218.

Hamdy A. Taha (2007): Operations Research, an introduction. (Pesarson Publication).

Kuhn H.W. (1955): The Hungarian Method for the Assignment Problem, Naval Research Loqistics Quarterly, **2**,83-97.

Bazarra M.S., Jarvis J. J. and Sherali H.D. (2005): Linear programming and network flows, Wiley.

Munkres, J. (1957): Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*. **5**,32-38.