A New Optimality Criteria for Efficiently Solving an Assignment Problems in Combinatorial Optimization

Sahil Ahuja, Anu Kathuria

The Technological Institute of Textile and Sciences, Bhiwani-127021
Email: *anu_sept24[at]rediffmail.com*

Abstract: ***The assignment problem, a fundamental combinatorial optimization operations research, optimally assigning n n tasks to n resources. While efficient algorithm exist such as the one proposed by Harold Kuhn in 1955. This paper presents a novel optimality criterion that requires fewer iterations, thus increasing computational efficiency. This proposed method is a systematic procedure, applicable to all types of assignment problems regardless of whether the objective function is to be maximized or minimized.***

Keywords: Assignment problem, Opportunity Cost, Combinatorial Optimization, Operations Research, Mathematical Modeling, Algorithm Efficiency

1. Introduction

An important topic, put forward immediately after the transportation problem, it the assignment problem. This is particularly important in the theory of decision making Different methods have been presented for assignment problem and various articles have been published on the subject. A considerable number of methods have been so far presented for assignment problem in which the Hungarian method in more convenient method among them. This iterative method is based on add or subtract a constant to every element of a row or column of the cost matrix, in a minimization model and create some zeros in the given cost matrix and then try to find a complete assignment in terms of zeros. By a complete assignment for a cost matrix $n \times n$, we mean an assignment plan containing exactly n assigned independent zeros, one in each row and one in each column. The main concept of assignment problem is to find the optimum allocation of a number of resources to in equal number of demand points. An assignment plan is optimal if optimizes the total cost or effectiveness of doing all the jobs. This paper attempts to propose a method for solving assignment problem which is different from the preceding methods.

1.1 Mathematical formulation of assignment problem

Mathematically an assignment problem can be stated as follows:

Optimize

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,\ldots,n$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,\ldots,n$$

$$x_{ij} = 0 \text{ or } 1,\ i = 1,\ldots,n,\ \ j = 1,\ldots,n.$$

where c_{ij} is the cost or effectiveness of assigning i^{th} job to j^{th} machine, and x_{ij} is to be some positive integer or zero, and the only possible integer is one, so the condition of $x_{ij} = 0$ or 1, is automatically satisfied. Associated to each assignment problem there is a matrix called cost or effectiveness matrix $\left[c_{ij}\right]$ where c_{ij} is the cost of assigning i^{th} job to j^{th} facility. In this paper we call it assignment matrix, and represent it as follows:

$$\begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ n \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \cdots & n \end{array} \\ \begin{pmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1n} \\ c_{21} & c_{22} & c_{23} & \cdots & c_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & c_{n3} & \cdots & c_{nn} \end{pmatrix} \end{array}$$

which is always a square matrix, thus each task can be assigned to only one machine. In fact any solution of this assignment problem will contain exactly m non-zero positive individual allocations. A customary and convenient method, termed as assignment algorithm" has been developed for such problems. This iterative method is known as Hungarian assignment method. It is based on add or subtract a constant to every element of a row or column of the cost matrix in a minimization model, and create some zeros in the given cost matrix and then try to find a complete assignment in terms of zeros.

2. A New Approach for Solving an Assignment Problem

This section presents a new method to solve the assignment problem which is different from the preceding. We call it "optimality criteria method." This method gives us an efficient method of finding optimal solution. This method takes into account not only the least cost c_{ij} but also the cost that just exceeds c_{ij}. Now, consider the assignment matrix where c_{ij} is the cost or effectiveness of assigning i^{th} job to j^{th} machine.

$$\begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ n \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \cdots & n \end{array} \\ \begin{pmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1n} \\ c_{21} & c_{22} & c_{23} & \cdots & c_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & c_{n3} & \cdots & c_{nn} \end{pmatrix} \end{array}$$

The new algorithm is as follows:

Step1. Determine the cost table from the given problem,

a) If the number of source is equal to the number of destination, go to step 3.
b) If the number of source is not equal to the number of destination, go to Step 2.

Step 2. Add a dummy source or dummy destination, so that the cost table becomes a square matrix. The cost entries of dummy source/destination are always zero.

Step 3. For each row of the assignment table identify the smallest and next-to-smallest costs. Determine the difference between them for each row; Display them alongside the assignment table by enclosing them in parenthesis against the respective rows. Similarly. Compute the differences for each column.

Step 4. Identify the row or column with the largest difference among all rows and columns.

a) If the greatest difference corresponds to i^{th} row and let c_{ij} be the smallest cost, in the i^{th} row. Circle that smallest cost and cross off its corresponding row and column.
b) If the greatest difference corresponds to j^{th} column and let c_{ij} be the smallest cost, in the j^{th} column. Circle that smallest cost and cross off its corresponding row and column.
c) In case if the occurs, then select that corresponding row or column which possesses smallest cost. Circle that smallest cost and cross off its corresponding row and column.

Step 5: Recompute the column and row differences for the reduced assignment table and go to Step 4.

Step 6: To find an optimal assignment

Special Case. Also, if even against that smallest cost there is tie. Then any row or column can be selected.

3. Numerical Examples

The following examples may be helpful to clarify the proposed method.

Example1. Consider the following assignment problem. Assign the five jobs to the five machines so as to minimize the total cost.

	1	2	3	4	5
1	12	8	7	15	4
2	7	9	1	14	10
3	9	6	12	6	7
4	7	6	14	6	10
5	9	6	12	10	6

Table 1

For each row of the assignment table identify the smallest and next-to-smallest costs. Determine the difference between them for each row. Display them alongside the assignment table by enclosing them in parenthesis against the respective rows. Similarly, compute the differences for each column.

	1	2	3	4	5	Difference
1	12	8	7	15	4	(3)
2	7	9	1	14	10	(6)
3	9	6	12	6	7	(0)
4	7	6	14	6	10	(0)
5	9	6	12	10	6	(0)
	(0)	(0)	(6)	(0)	(2)	

Table 2

Identify the row or column with the largest difference among all rows and columns. Since there is a tie between the greatest differences that we have computed so. Select that corresponding row or column which possesses smallest cost. Circle that smallest cost and cross off its corresponding row and column.

	1	2	3	4	5	
1	12	8	7	(15)	4	(3)
2	7	9	1	14	10	(6)
3	9	6	12	6	7	(0)
4	7	6	14	6	10	(0)
5	9	6	12	10	6	(0)
	(0)	(0)	(6)	(0)	(2)	

Table 3

The reduced matrix will be,

	1	2	4	5	
1	12	8	(15)	4	(4)
3	9	6	6	7	(0)
4	7	6	6	10	(0)
5	9	6	10	6	(0)
	(2)	(0)	(0)	(2)	

Table 4

Now, the greatest difference is 4, so among the first row, the smallest cost is 4. So, circle that smallest cost and cross off its corresponding row and column.

	1	2	4	5	
1	12	8	15	④	(4)
3	9	6	6	7	(0)
4	7	6	6	10	(0)
5	9	6	10	6	(0)
	(2)	(0)	(0)	(2)	

Table 5

The reduced matrix will be:

	1	2	4	
3	9	6	6	(0)
4	7	6	6	(0)
5	9	6	10	(3)
	(2)	(0)	(0)	

Table 7

Now, the greatest difference is 3, so among the fifth row, the smallest cost is 6. So circle that smallest cost and cross off its corresponding row and column.

	1	2	4	
3	9	6	6	(0)
4	7	6	6	(0)
5	9	⑥	10	(3)
	(2)	(0)	(0)	

Table 8

The reduced matrix will be:

	1	4	
3	9	6	(3)
4	7	6	(1)
	(2)	(0)	

Now, the greatest difference is 3, so among the third row, the smallest cost is 6. So circle that smallest cost and cross off its corresponding row and column.

	1	4	
3	9	⑥	(3)
4	7	6	(1)
	(2)	(0)	

Now, also encircle the remaining cost 7.

So, the complete assignment is

	1	2	3	4	5
1	2	8	7	15	④
2	7	9	①	14	10
3	9	6	12	⑥	7
4	⑦	6	14	6	10
5	9	⑥	12	10	6

Determine particular sub matrix of order 2 against first allocation consisting of a least one allocation diagonally

Justification:

$A[i, j] =$

	1	2	3	4	5
1	2	8	7	15	④
2	7	9	①	14	10
3	9	6	12	⑥	7
4	⑦	6	14	6	10
5	9	⑥	12	10	6

Table 9

In the above matrix $A[i, j]_{5\times 5}$,

For the first and second row entries 4 (in first row) and 1 (in second row) are allocated diagonally i.e.

	3	5
1	7	④
2	①	10

Step (a) Take Sum (4, 1). If Sum (4, 1) ≤ Sum (7, 10), then move to step 2 otherwise interchange 4 & 1 with 7 & 10 respectively.

Step (b) Find another sub matrix of order 2 × 2 consisting of single allocation that must be lying on the right side of element 4. If no, then the existing cell will give optimal cell. Similarly, take the matrix of order 2 for the second and third row. Here entries allocated diagonally are 1 and 6 i.e.

	3	4
2	①	14
3	12	⑥

Step (a) Take Sum (1, 6). If Sum (1, 6) ≤ Sum (14, 12), then move to step 2 otherwise interchange 1 & 6 with 14 & 12 respectively.

Step (b) Find another sub matrix of order 2 ×2 consisting of single allocation that must be lying on the right side of element 1 i.e.

	4	5
2	14	10
3	⑥	7

Step (c) Since Sum (10, 6) ≤ Sum (14, 7), the existing cell will give optimal cell.

Now take the matrix of order 2 for the third and fourth row. Here entries allocated diagonally are 6 and 7 i.e.

$$\begin{array}{c} \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{matrix} 1 & 4 \end{matrix} \\ \begin{pmatrix} 9 & 6 \\ 7 & 6 \end{pmatrix} \end{array}$$

Step (a) Take Sum (6, 7). If Sum (6, 7) ≤ Sum (9, 6), then move to step 2 otherwise interchange 6 & 7 with 9 & 16 respectively.

Step (b) Find another sub matrix of order 2 × 2 consisting of single allocation that must be lying on the right side of element 6. If no, then the existing cell will give optimal cell. Similarly take the matrix of order 2 for the fourth and fifth row. Here entries allocated diagonally are 7 and 6 i.e.

$$\begin{array}{c} \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{matrix} 1 & 2 \end{matrix} \\ \begin{pmatrix} 7 & 6 \\ 9 & 6 \end{pmatrix} \end{array}$$

Step (a) Take Sum (7, 6). If Sum (7, 6) ≤ Sum (6, 9), then move to step 2 otherwise interchange 7 & 6 with 6 & 9 respectively.

Step (b) Find another sub matrix of order 2 × 2 consisting of single allocation that must be lying on the right side of element 7.

$$\begin{array}{c} \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{matrix} 2 & 3 \end{matrix} \\ \begin{pmatrix} 6 & 14 \\ 6 & 12 \end{pmatrix} \end{array}$$

Step (c). Here Sum (6, 14) > Sum (6, 12). So interchanging is maximum i.e. on interchange 7 & 6 with 6 & 12 then element affected is 1 (minimum) in second row. So these entries cannot be interchanged.

4.1 Another way for finding an optimal solution

City	A	B	C	D
A	5	3	1	8
B	7	9	2	6
C	6	4	5	7
D	5	7	7	6

In the above matrix first of all we will find the feasible solution.
For each row of the assignment table identify the smallest and next-to-smallest costs. Determine the difference between them for each row. Display them alongside the assignment table by enclosing them in parenthesis against the respective rows. Similarly, compute the differences for each column.

City	A	B	C	D	
A	5	3	1	8	(2)
B	7	9	2	6	(4)
C	6	4	5	7	(1)
D	5	7	7	6	(1)
	(0)	(1)	(1)	(0)	

Identify the row or column with the largest difference among all rows and columns. The greatest difference is 4, so among the second row, the smallest cost is 2.Circle that smallest cost and cross off its corresponding row and column.

City	A	B	C	D	
A	5	3	1	8	(4)
B	7	9	2	6	(1)
C	6	4	5	7	(1)
D	5	7	7	6	(1)
	(1)	(0)			

The reduced matrix will be:

City	A	B	D	
A	5	3	8	(2)
C	6	4	7	(2)
D	5	7	6	**(1)**
	(1)	(0)	(1)	(0)

Since there is a tie between the greatest difference that we have computed so. Select that corresponding row or column which possesses smallest cost. Circle that smallest cost and cross off its corresponding row and column.

City	A	B	D	
A	5	3	8	(2)
C	6	4	7	(2)
D	5	7	6	(1)
	(1)	(0)	(1)	(0)

The reduced matrix will be:

City	A	D	
C	6	7	(1)
D	5	6	(1)
	(1)	(1)	

Since there is a tie between the greatest difference that we have computed so. Select that corresponding row or column which possesses smallest cost. Circle that smallest cost and cross off its corresponding row and column.

City	A	D	
C	6	7	(1)
D	5	6	(1)

Now, also encircle the remaining cost 7.
So, the complete assignment is

City	A	B	C	D
A	5	3	1	8
B	7	9	2	6
C	6	4	5	7
D	5	7	7	6

Next step is to find an optimal solution.

Here allocated elements are termed as key elements and X means no route.

Step 1. In the above matrix for every allocation corresponding to each row of the matrix determine all the elements smaller to itself and only one element greater to itself.

That is, in the first row for key element 3, the smaller element is 1 and the greater element is 5.

So there will be 3 elements defining the path i.e. 1, 3 and 5. Similarly, in the second row for key element 2, the smaller element is 2 itself and greater element is 6. So there will be 2 elements i.e. 2, 6.

Similarly, in the third row for key element 7, the smaller elements are 4, 5, 6, 7 and greater element is 7 itself. So there will be 4 elements i.e. 4, 5, 6, 7.

Similarly, in the fourth row for key element 5, the smaller element is 5 itself and greater element is 6. So there will be 2 elements i.e. 5, 6.

Step 2. If minimum number of elements of any row is equal to the next minimum number of elements for any other row, then go to step 3 otherwise determine another element greater to key element for the row containing minimum number of elements.

Here the minimum number of element is 2 and the next minimum number of element is also 2. So, move to step 3.

Step 3. So in the first row the 3 elements are 1, 3 and 5.

(a) Starting with first element i.e. 1.
From 1, we can move to either 2 or 6. But we cannot move to 2 because 2 is below 1 (an allocated cell).
So the next element is 6.
From 6, we move to either 4 or 5 or 6 or 7. But we cannot move to 5, 7 because 5 is below 1 and 7 is below 6. So next element can be 4 or 6.

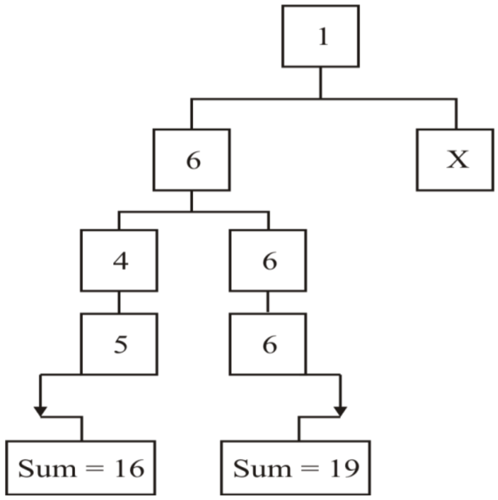
(i) If the next element is 4.
From 4, we can move to either 5 or 6. But we cannot go to 6 because 6 is below the element 6 in the second row.

So the final allocation is 1 (in the first row), 6 (in the second row), 4 (in the third row) and 5 (in the fourth row). Sum of these key elements is 16.

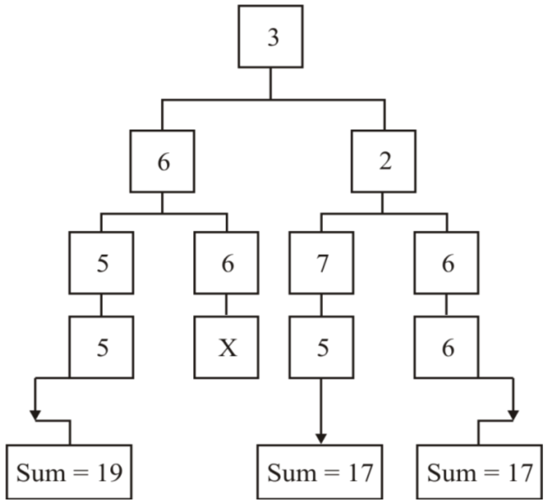
(ii) If the next element is 6.
From 6, we can move to either 5 or 6. But we cannot go to 5 because 5 is below the element 6 in the third row.

So the final allocation is 1 (in the first row), 6 (in the second row), 6 (in the third row) and 6 (in the fourth row). Sum of these key elements is 19.

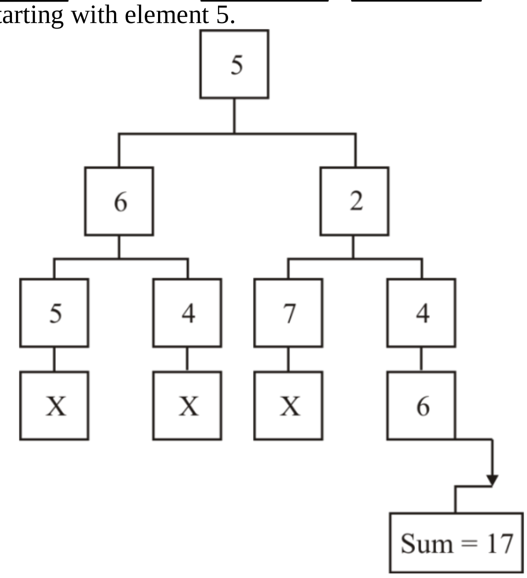
The above step 3 can become clearer from the following chart:



(b) Starting with element 3.



(c) Starting with element 5.



Among these three, the minimum sum is 16 and the path is:
A → C, B → D, C→ B, D → A.

4. Conclusion

This paper, introduced a fast and simple method for solving assignment problems, applicable to both maximization and minimization of objective functions. The proposed method requires fewer iterations than existing methods enhancing computational efficiency. Our future work will focus on developing an algorithm based on this method.

References

[1] Barr, R., Glover, F., and Klingman, D., 1977. *"*The Alternating Basis Algorithm for Assignment Problems,*"* Math. Programming, Vol. 13, pp. 1-13.
[2] Croes, G.A. (1958): 'A method for solving travelling salesman problem', Operations Research, 6, pp. 791-812.
[3] Dantzig. G.B. Fulkerson, D.R. and Jhonson, S.M.(1954) 'Solutions of a large scale Travelling salesman Problem', Operations Research, 2, pp. 393-410.
[4] Flood, M.M. (1956): The travelling salesman problem', Oper. Res., 4, pp. 61-75.
[5] James Munkres (1957): An algorithm for the transportation problem', Journal of the Society for industrial and Applied Mathematics, Vol. 5, No. 1. 32-38.
[6] Little JD.C, Murthy, K.G, Sweeny, D W., and Karel, C. (1963): An algorithm for the travelling salesman problem', Operations Research,pp.979-989.
[7] Rao, S.S. (1984): Optimisation: Theory and applications, 2nd edition Wiley Eastern Limited.
[8] Taha, H.A. (1976): Operations Research—An Introduction on 2/e, New York: Macmillan.
[9] Wagner, HM. (1974): Principles of Operation Research, New Delhi: Prentice-Hall of India.