American Journal of Operations Research, 2016, 6, 81-89
Published Online January 2016 in SciRes. http://www.scirp.org/journal/ajor
http://dx.doi.org/10.4236/ajor.2016.61011

A New Approach of Solving Single Objective Unbalanced Assignment Problem

Ventepaka Yadaiah[1], **V. V. Haragopal**[2]

[1]Department of Mathematics, Osmania University, Hyderabad, India
[2]Department of Statistics, Osmania University, Hyderabad, India
Email: v.yadaiah@gmail.com, haragopalvajjha@gmail.com

Received 9 December 2015; accepted 24 January 2016; published 29 January 2016

Abstract

In this paper, we discuss a new approach for solving an unbalanced assignment problem. A Lexi-search algorithm is used to assign all the jobs to machines optimally. The results of new approach are compared with existing approaches, and this approach outperforms other methods. Finally, numerical example (Table 1) has been given to show the efficiency of the proposed methodology.

Keywords

Assignment Problem, Lexi-Search Algorithm, Jobs Clubbing Method

1. Introduction

Consider a problem which consists of a set of "n" machines $M=\{M_1,M_2,M_3,\cdots,M_n\}$. A set of "m" jobs $J=\{J_1,J_2,J_3,\cdots,J_m\}$ which are to be considered assign for execution on "n" available machines. The execution cost of each job on all the machines are known and mentioned in the matrix, namely assigned cost matrix (ACM) of order, where $m>n$. The unbalanced assignment problem is a special type of linear programming problem in which our objective is to assign number of salesmen to number of areas at a minimum cost (time). The mathematical formulation of the problem suggests that this is a 0 - 1 programming problem. It is highly degenerate all the algorithms developed to find optimal solution of transportation problem, applicable to unbalanced assignment problem. However, due to its highly degeneracy nature a specially designed algorithm, widely known as Hungarian method proposed by Kuhn [1], is used for its solution, and Kadhirvel and Balamurugan [2] solved the unbalanced assignment problems using triangular fuzzy Numbers. Different methods have been presented for Assignment Problem and various articles have been published on the subject [3]-[7].

The objectives are to determine the optimal assignment cost, in such a way that all the jobs are to be allotted on the available machines in an optimum way. The mathematical formulation of the assignment problem [8] [9] is as follows.

How to cite this paper: Yadaiah, V. and Haragopal, V.V. (2016) A New Approach of Solving Single Objective Unbalanced Assignment Problem. *American Journal of Operations Research*, **6**, 81-89. http://dx.doi.org/10.4236/ajor.2016.61011

Table 1. Assigned cost matrix (ACM).

	M_1	M_2	M_3	M_4	M_5
J_1	300	290	280	290	210
J_2	250	310	290	300	200
J_3	180	190	300	190	180
J_4	320	180	190	240	170
J_5	270	210	190	250	160
J_6	190	200	220	190	140
J_7	220	300	230	180	160
J_8	260	190	260	210	180

2. Model Construction of Simple Assignment Problem

Minimize (Maximize):

$$Z = \sum_{i=1}^{m}\sum_{i=1}^{n} C_{ij} X_{ij}$$

Subject to

$$\sum_{j=1}^{n} X_{ij} = 1; \text{ for } i = 1, 2, 3, \cdots, m$$

$$\sum_{i=1}^{m} X_{ij} = 1; \text{ for } j = 1, 2, 3, \cdots, n$$

where $X_{ij} = \begin{cases} 1, & \text{if the } i^{th} \text{ job is assigned to the } j^{th} \text{ machine.} \\ 0, & \text{if the } i^{th} \text{ job is not assigned to the } j^{th} \text{ machine.} \end{cases}$

Problem definition:

Also, if the numbers of jobs are not equal to number of machines, then it is known as an unbalanced assignment problem. Now consider the assumptions of choosing an unbalanced assignment problem as:

- The completion of a program from computational point of view means that the all jobs are assigned to various machines and final optimal assignment cost has been obtained.
- The number of jobs are more than number of machines.

The variants of assignment problem are considered by various researchers like Kagade & Bajaj [10] and Avanish Kumar [11]. From the work of these authors, they found that the approach of clubbing the costs of the jobs was implemented for multi objective problems and single objective problems, where as this paper considers the clubbing of jobs for an assignment problem by the exact solution problem with Lexi-search approach [12] [13].

3. Methodology

To determine the assignment cost as well as combination of job (s) Vs machine (s) of an unbalanced assignment problem for a set of “n” machines $M = \{M_1, M_2, M_3, \cdots, M_n\}$. A set of “m” jobs $J = \{J_1, J_2, J_3, \cdots, J_m\}$ which are to be considered as assigned for execution on “n” available machines with an execution cost C_{ij}, where $i = 1, 2, \cdots, m$ and $j = 1, 2, \cdots, n$ are mentioned in the ACM of order, where $m > n$. First of all, we obtain the sum of each row and each column of the ACM store and the results should be arranged in the array, namely, Sum_Row and Sum_Column. Then we select the first m rows (jobs) on the basis of Sum_Row that is, starting with the most minimum to next minimum to the array Sum_Row and deleting rows (jobs) correspo- nding to the remaining (m-n) jobs. Store results in the new array that will be the array for the first sub problem (**Table 2**). Repeating this process until the remaining jobs become less than “n” machines, when remaining jobs are less than n then deleting (n-m) columns (machines) on the basis of Sum_Column. That is, corresponding to value (s) most maximum to next maximum to form the last sub problem (**Table 3**). Store the results in the new array that shall be the array for the last sub problem. which are now balanced assignment problems, in this way for the

Table 2. First sub-problem $N_1ACM(,)$.

	M_1	M_2	M_3	M_4	M_5
J_3	180	190	300	190	180
J_4	320	180	190	240	170
J_5	270	210	190	250	160
J_6	190	200	220	190	140
J_7	220	300	230	180	160

Table 3. Second sub problem $N_2ACM(,)$.

	M_2	M_4	M_5
J_1	290	290	210
J_2	310	300	200
J_8	190	210	180

defined assignment problem.Now we apply the Lexi-search approach to obtain the exact optimum solution of each sub problem (**Tables 4-7**). Finally, add the total assignment cost of each sub problem to obtain the optimal assignment cost along with assignment sets. And also we check the assignment cost for jobs clubbing problem (**Table 8**) through Lexi-search approach (**Tables 9-11**), getting the same value. To solve this problem we follow the following algorithm.

Algorithm

Step-1: Consider "m" jobs on "n" machines costs given as a matrix (ACM), which is an unbalanced assignment problem where $m > n$.

Step-2:

Step-2.1: Obtain the sum of each row and column of the ACM and the store the results in the arrays namely Sum_Row and Sum_Column.

Step-2.2: Select the first m rows (jobs) on the basis of Sum_Row. That is, starting with the most minimum to next minimum to the array Sum_Row and deleting rows (jobs) corresponding to the remaining (m-n) jobs. Store the results in the new array that shall be the array for the first sub problem.

Step-2.2.1: If there is no remaining jobs, *i.e.*, (m-n = 0), then go to step-3.

Step-2.2.2: If the remaining (m-n) jobs are still more than n, then repeat step-2.2 for the remaining jobs to form next sub-problem (s), else, step-2.3.

Step-2.3: If remaining jobs are less than n then deleting (n-m) columns (machines) on the basis of Sum_Column. That is corresponding to value (s) most maximum to next maximum to form the last sub problem. Store the results in the new array that shall be the array for the last sub problem.

Step-3: If the total effectiveness of ACM is to be maximized, change the sign of each cost element in the effectiveness matrix and go to step-4, otherwise go directly to step-5 if ACM has the total value as minimum.

Step-4: Arrange all the jobs $J_1, J_2, J_3, \cdots, J_n$ according to their cost (*i.e.* available jobs). This arrangement consists of n columns and m rows. Each column represents a machine, and the elements in that column are the costs arranged in increasing order according to their jobs.

Step-5: Include the job from the first machine in the partial solution value (psv) "w". If the cost itself is greater than or equal to trial value (TRV) then stop. Otherwise go to next step.

Step-6: Calculate the bound.

Step-7: If the sum of bound and psv is greater than or equal to TRV then drop the job added in step 5, and go to step 5. Otherwise go to next step, *i.e.* go to Sub block (GS).

Step-8: Include the next available job (from the last job included in the partial solution "w") into the partial solution.

Table 4. Alphabet table $N_1ACM(,)$.

M_1	M_2	M_3	M_4	M_5
J_3-180	J_4-180	J_4-190	J_7-180	J_6-140
J_6-190	J_3-190	J_5-190	J_3-190	J_5-160
J_7-220	J_6-200	J_6-220	J_6-190	J_7-160
J_5-270	J_5-210	J_7-230	J_4-240	J_4-170
J_4-320	J_7-300	J_3-300	J_5-250	J_3-180

Table 5. Search table $N_1ACM(,)$.

M_1	M_2	M_3	M_4	M_5	A.B	Remark
J_3-180	J_4-180	$J_4-190RP$ J_5-190	J_7-180	J_6-140		
Bound = 870 (180 + 180 + 190 + 180 + 140)	Bound = 870 (180 + 180 + 190 + 180 + 140)	Bound = 870: (180 + 180 + 190 +180 + 140)	Bound = 870 (180 + 180 + 190 + 180 + 140)	Bound = 870: (180 + 180 + 190 + 180 + 140)	870	
			$J_3-190RP$ J_6-190 Bound: 900 > 870 (180 + 180 + 190 + 190 + 160)			GNSB
		J_6-220 Bound: 920 > 870 (180 + 180 + 220 +180 + 160)				GNSB
	$J_3-190RP$ J_6-200 Bound: 910 > 870 (180 + 200 + 190 + 180 + 160)					GNSB
J_6-190 Bound: 920 > 870 (190 + 180 + 190 + 180 + 180)						GNSB
J_7-220 Bound: 920 > 870 (220 + 180 + 190 + 190 + 140)						
J_5-270 Bound: 1030 > 870 (270 + 180 + 220 + 180 + 180)						
J_4-320 Bound: 1020 > 870 (320 + 190 + 190 + 180 + 140)						

Table 6. Alphabet table $N_2ACM(,)$.

M_2	M_4	M_5
J_8-190	J_8-210	J_8-180
J_1-290	J_1-290	J_2-200
J_2-310	J_2-300	J_1-190

Table 7. Search table $N_2ACM(,)$.

M_2	M_4	M_5	A.B	Remark
J_8-190	$J_8-210RP$	$J_8-180RP$		
Bound: 680	J_1-290	J_2-200	680	
(190 + 290 + 200)	Bound: 680	Bound: 680		
	(190 + 290 + 200)	(190 + 290 + 200)		
	J_2-300			
	Bound: 700 > 680			GNSB
	(190 + 300 + 210)			
J_1-290				
Bound: 700 > 680				
(290 + 210 + 200)				
J_2-310				
Bound: 730 > 680				
(310 + 210 + 210)				

Table 8. Jobs clubbing modified problem is $N_3ACM(,)$.

	M_1	M_2	M_3	M_4	M_5
J_1*J_7	520	590	510	470	370
J_2*J_6	440	510	510	490	340
J_2	180	190	300	190	180
J_4*J_8	580	370	450	450	350
J_5	270	210	190	250	160

Table 9. Alphabet table $N_3ACM(,)$.

M_1	M_2	M_3	M_4	M_5
J_3-180	J_3-190	J_5-190	J_3-190	J_5-160
J_5-270	J_5-210	J_3-300	J_5-250	J_3-180
J_2*J_6-440	J_4*J_8-370	J_4*J_8-450	J_4*J_8-450	J_2*J_6-340
J_1*J_7-520	J_2*J_6-510	J_1*J_7-510	J_1*J_7-470	J_4*J_8-350
J_4*J_8-580	J_1*J_7-590	J_2*J_6-510	J_2*J_6-510	J_1*J_7-370

Table 10. Search table $N_3ACM(,)$.

M_1	M_2	M_3	M_4	M_5	A.B	Remark
J_3-180 Bound: 1650 (180 + 210 + 450 + 470 + 340)	$J_3-190RP$ J_5-210 Bound: 1650 (180 + 210 + 450 + 470 + 340)	$J_5-210RP$ $J_3-190RP$ J_4*J_8-450 Bound: 1650 (180 + 210 + 450 + 470 + 340)	$J_3-190RP$ $J_5-250RP$ $J_4*J_8-450RP$ J_1*J_7-470 Bound: 1650 180 + 210 + 450 + 470 + 340)	$J_5-210RP$ $J_3-190RP$ J_2*J_6-340 Bound: 1650 (180 + 210 + 450 + 470 + 340)	1650	JB
				$J_4*J_8-350RP$ $J_1*J_7-370RP$		GNSB
			J_2*J_6-490 Bound: 1700 > 1650 (180 + 210 + 450 + 490 + 370)			GNSB
		J_1*J_7-510 Bound: 1690 > 1650 (180 + 210 + 510 + 450 + 340)				JB
		J_2*J_6-510 Bound: 1720 > 1650 (180 + 210 + 510 + 450 + 370)				GNSB
	J_4*J_8-370 Bound: 1550 < 1650 (180 + 370 + 190 + 470 + 340)	J_5-190 Bound: 1550 < 1650 (180 + 370 + 190 + 470 + 340)	$J_5-190RP$ $J_3-300RP$ $J_4*J_8-450RP$ J_1*J_7-510 Bound: 1550 < 1650 (180 + 370 + 190 + 470 + 340)	$J_5-160RP$ $J_3-180RP$ J_2*J_6-340 Bound: 1550 < 1650 (180 + 370 + 190 + 470 + 340)	1550	JB
				$J_4*J_8-350RP$ $J_1*J_7-370RP$		GNSB
			J_2*J_6-490 Bound: 1600 > 1550 (180 + 370 + 190 + 490 + 370)			GNSB

Step-9: If partial solution value is greater than or equal to the TRV then drop the job added in step-8, and go to step-7. Otherwise go to step-10.

Step-10: If the sum of bound and psv is greater than or equal to TRV then drop the newly added job in step-8, and go to step -7.otherwise go to step 11.

Step-11: If the partial solution contains n − 1 jobs add the dummy job to the partial solution if it is greater than or equal to TRV then drop the dummy job and last two jobs from the partial solution. That is Jump out to the next higher order blocks (JO). If "w" contains only one job, go to step-5, otherwise go to step-8. Otherwise go to the next step.

Step-12: Now calculate the bound.

Step-13: If the sum of bound and psv is greater than or equal to TRV then drop the dummy job and also last job from "w", and go to step-8. Otherwise go to step-14.

Table 11. Search table $N_3ACM(,)$.

M_1	M_2	M_3	M_4	M_5	A.B	Remark
		$J_3-300RP$				JB
		$J_4*J_8-450RP$				
		J_1*J_7-510				
		Bound: 1650 > 1550				
		(180 + 370 + 510 + 250 + 340)				
		$J_3-300RP$				GNSB
		$J_4*J_8-450RP$				
		J_2*J_6-510				
		Bound: 1680 > 1550				
		(180 + 370 + 510 + 250 + 370)				
	J_2*J_6-510					
	Bound: 1700 > 1550					
	(180 + 510 + 190 + 450 + 370)					
	J_1*J_7-590					
	Bound: 1750 > 1550					
	(180 + 590 + 190 + 450 + 340)					
J_5-270						
Bound: 1720 > 1550						
(270 + 190 + 450 + 470 + 340)						
J_2*J_6-440						
Bound: 1640 > 1550						
(440 + 190 + 190 + 450 + 370)						
J_1*J_7-520						
Bound: 1690 > 1550						
(520 + 190 + 190 + 450 + 340)						
J_4*J_8-580						
Bound: 1770 > 1550						
(580 + 190 + 190 + 470 + 340)						

Step-14: Include the latest possible job from the dummy job in“w”

Step-15: If psv is greater than or equal to TRV then drop the last dummy job and also the job from which the i^{th} dummy job was assigned, and go to step-8. Otherwise go to next step.

Step-16: Now calculate the bound.

Step-17: If sum of bound and psv is greater than or equal to TRV then drop the recently added job in “w” and go to step-14. Otherwise go to next step.

Step-18: Include the latest available job from the last job in “w”

Step-19: Now calculate the bound.

Step-20: If the sum of bound and psv is greater than or equal to TRV then drop the latest job, and go to step-18. Otherwise go to next step.

Step-21: If the number of elements in “w” is less than “n” go to step-18. Otherwise go to next step.

Step-22: Replace TRV by partial solution value and trial solution by w. Now go to step-18.

4. Illustration

A company is faced with the problem of assigning five different machines to eight different jobs (**Table 1**). The costs are estimated as follows (in hundreds of rupees):

Solve the problem assuming that the objective is to minimize the total cost. Now obtain the sum of each row and column of $ACM(,)$, *i.e.*, the sum of each row and each column is as follows:

$$Sum\_Row = \begin{array}{cccccccc} J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 & J_8 \\ 1370 & 1350 & 1040 & 1100 & 1080 & 0940 & 1090 & 1100 \end{array}$$

$$Sum\_Column = \begin{array}{ccccc} M_1 & M_2 & M_3 & M_4 & M_5 \\ 1990 & 1870 & 1960 & 1850 & 1400 \end{array}$$

We partition the matrix $ACM(,)$ to define the first sub-problem $N_1ACM(,)$ by selecting rows corresponding J_3, J_4, J_5, J_6, J_7 and second sub problem $N_2ACM(,)$ by selecting rows corresponding to the jobs J_1, J_2, J_8 and by deleting columns corresponding to M_1, M_3, then the modified matrices are as follows:

Sub-Problem-I: $N_1ACM(,)$

Sub-Problem-II: $N_2ACM(,)$

4.1. Now apply the Lexi-search method for Sub-Problem-I: $N_1ACM(,)$

RP: Repitition, GNSB or JB: Go to next super block (JB), A.B or TRV = Absolute bound or Trail bound.

The final optimal assignments of $N_1ACM(,)$ as follows:

$J_3 \to M_1$, $J_4 \to M_2$, $J_5 \to M_3$, $J_6 \to M_5$, $J_7 \to M_4$

4.2. Now Apply the Lexi-Search Method for Sub-Problem-II: $N_2ACM(,)$

The final optimal assignments $N_2ACM(,)$ is: $J_1 \to M_4$, $J_2 \to M_5$, $J_8 \to M_2$

The final optimal assignments assigned cost matrix (ACM) is : $J_1 \to M_4$, $J_2 \to M_5$, $J_3 \to M_1$, $J_4 \to M_2$, $J_5 \to M_3$, $J_6 \to M_5$, $J_7 \to M_4$, $J_8 \to M_2$. The Hungarian method gives us total assignment cost as 890 along with the other one job assigned to dummy machine, in other words the job that is assigned to dummy machine under the Hungarian method was ignored for further processing. While, the original problem was divided into two sub problems, which are balanced assignment problem in nature. Now for the two sub problems with the use of Lexi-search approach, the total cost 870 is recorded for the sub problem-I along with none of the jobs assigned to dummy machine, and the total cost 680 was recorded for the second sub problem-II along with none of the jobs assigned to dummy machine.Now the total cost of the assigned cost matrix (ACM) is 870 + 670 = 1550.

5. Job Clubbing Method

Jobs Clubbing Modified Problem Is $N_3ACM(,)$: Lexi-Search Approach

The final optimal assignments $N_3ACM(,)$ as follows: $J_3 \to M_1$, $J_4 * J_8 \to M_2$, $J_5 \to M_3$, $J_1 * J_7 \to M_4$, $J_2 * J_6 \to M_5$

Total assignment cost = 1550.

Problem	Hungarian method	Lexi-search method
Unbalanced assignment problem	Uses the dummy assignment	Never uses the dummy assignment
Jobclubbing	Gives optimum	Gives exact optimum

6. Conclusion

The above illustration was taken by the defined algorithm and implemented on several sizes of the problems to test the effectiveness of the algorithm. This approach was implemented on different sizes of unbalanced assignment problems. From the above, we notice that the standard Hungarian method uses the dummy assignment which may not be possible in some applications, whereas this new approach never assigns the dummy machine in getting the optimum value. The time complexity with the Lexi-search method is verified and found that they are the same in getting optimum. Here, the optimum value of the original unbalanced assignment problems varies from that of balanced assignment problems either in Hungarian method or Lexi-search approach. The only advantage is that the Lexi-search method gives an exact optimum value with the same time complexity. Therefore the present paper suggests a new approach of clubbing the jobs for solving the unbalanced assignment problem with Lexi-search methodology.

References

[1] Kuhn, H.W. (1955) The Hungarian Method for the Assignment Problem. *Naval Research Logistic Quarterly*, **2**, 83-97. http://dx.doi.org/10.1002/nav.3800020109

[2] Kadhirvel, K. and Balamurugan, K. (2013) Method for Solving Unbalanced Assignment Problems Using Triangular Fuzzy Numbers. *Journal of Engineering Research and Applications*, **3**, 359-363.

[3] Turkensteen, M., Ghosh, D., Goldengorin, B. and Sierksma, G. (2008) Tolerance-Based Branch and Bound Algorithm for the ATSP. *European Journal of Operational Research*, **189**, 775-788. http://dx.doi.org/10.1016/j.ejor.2006.10.062

[4] Basirzadeh, H. (2012) Ones Assignment Method for Solving Assignment Problems. *Applied Mathematical Sciences*, **6**, 2345-2355.

[5] Frank, A. (2004) On Kuhn's Hungarian Method—A Tribute from Hungary. Wiley Inter Science, Published Online.

[6] Pentico, D.W. (2007) Assignment Problem: A Golden Anniversary Survey. *European Journal of Operation Research*, **176**, 774-793. http://dx.doi.org/10.1016/j.ejor.2005.09.014

[7] Singh, S., Dubey, G.C. and Rajesh Shrivastava, R. (2012) A Comparative Analysis of Assignment Problem. *IOSR Journal of Engineering* (IOSRJEN), **2**, 1-15. http://dx.doi.org/10.9790/3021-02810115

[8] Gillett Billy, E. (2000) Introduction to Operations Research—A Computer Oriented Algorithm Approach. Tata McGraw Hill, New Delhi.

[9] Taha, H.A. (1971) Operation Research: An Introduction. MacMillan Inc., New York.

[10] Kagade, K.L. and Bajaj, V.H. (2010) A New Approach to Solve Fuzzy Multi-Objective Unbalanced Assignment Problem. *International Journal of Agriculture Statistics*, **6**, 31-40.

[11] Kumar, A. (2006) A Modified Method for Solving the Unbalanced Assignment Problems. *Journal of Applied Mathematics and Computation*, **176**, 76-82. http://dx.doi.org/10.1016/j.amc.2005.09.056

[12] Pandit, S.N.N. (1963) Some Quantitative Combinatorial Search Problems. PhD Thesis, IIT, Khargpur.

[13] Ramesh, M. (1997) Lexi-Search Approach to Some Combinatorial Programming Problem. PhD Thesis, University of Hyderabad, Hyderabad.