AN EFFECTIVE METHOD FOR SOLVING INTERVAL VALUED UNBALANCED ASSIGNMENT PROBLEM

P. Rajarajeswari[1] & D. Maheswari[2]

[1]Assistant Professor of Mathematics, [2]Associate Professor of Mathematics,

Chikkanna Government Arts College, Tamilnadu, India.
CMS College of Science and Commerce, Tamilnadu, India.
[1](p.rajarajeswari29@gmail.com)
[2](maheswari.jag@gmail.com)

Abstract: *In this paper, we discuss a new approach for solving an unbalanced interval valued assignment problem. A modified Hungarian algorithm is used to assign all the jobs to machines optimally. Here dummy machine concept is not used to find the optimum solution. Numerical examples are solved to show the efficiency of the proposed methodology.*

Keywords: Interval numbers, unbalanced problem, Assignment problem, modified Hungarian method.

1. INTRODUCTION

The assignment problem is a combinatorial optimization problem in the field of operations research. It is a special case and completely degenerate form of a transportation problem, which occurs when each supply is 1 and each demand is 1. It consists of assigning a number of tasks to an equal number of agents (each agent is assigned to exactly one task and each task has exactly one agent assigned to perform it) in such a way that the objective is either to minimize the total cost or to maximize the total profit.

Several authors have proposed different methods for solving the unbalanced assignment problem in which all jobs get executed. Kumar [9] proposed a modified method for solving unbalanced assignment problem where more than one jobs are assigned to a single machine. Further Yadaiah et al[10] applied Lexi search approach for solving unbalanced assignment problem and obtained same result as that of [9].In 2016, Betts [3] proposed an better approach than the approach of Yadaiah and Kumar and got better result also.

The methods suggested so far in the Hungarian method are based on the assumption that some of the jobs are assigned to a dummy or fictitious machine, which later may be ignored. The present paper suggests a modified method for solving unbalanced assignment problems. The method is capable of assigning all the jobs to machines optimally. The standard Hungarian method uses the dummy assignment which may not be possible in some applications, whereas this modified approach never assigns the dummy machine in getting the optimum value. The unbalanced assignment problem is a particular case of transportation problem in which our objective is to assign a number of jobs to an equal number of machines in order to minimize the cost or maximize profit. There exist several approaches that have been developed for finding an optimal policy of assigning jobs to machines.

In this paper we considered modified Hungarian method which was proposed by Rabbani [10]. In this method all the jobs are executed and the cost is minimized without considering dummy machines. Here Rabbani [10] problem is considered as an interval numbers and it is solved by using the proposed algorithm and obtained the optimum solution in terms of interval values.

II. PRELIMINARIES

Definition: 2.1

Let $\bar{a} = [a_L, a_R] = \{x \in \Re, a_L \le x \le a_R \ \&\ a_L, a_R \in \Re\}$ be an interval on the real line $\Re$. If $\bar{a} = a_L = a_R = a$ then $\bar{a} = [a,a] = a$ is a real number. The midpoint and half width of an interval number $\bar{a} = [a_L, a_R]$ is defined as $m(\bar{a}) = \left(\frac{a_L + a_R}{2}\right)$ and $w(\bar{a}) = \left(\frac{a_R - a_L}{2}\right)$. Now the interval number $\bar{a}$ can be expressed as $\bar{a} = < m(\bar{a}), w(\bar{a}) >$.

Definition: 2.2

Let $\le$ be an extended order relation between the interval numbers $\bar{a} = [a_L, a_R]$, $\bar{b} = [b_L, b_R]$ in IR, then for $m(\bar{a}) < m(\bar{b})$ we construct a premise $(\bar{a} < \bar{b})$ which means that $\bar{a}$ inferior to $\bar{b}$. An acceptability function $A_{\le} : IR \times IR \rightarrow [0,\infty)$ is defined as

$$A_{\le}(\bar{a}, \bar{b}) = A(\bar{a} \le \bar{b}) = \frac{m(\bar{b}) - m(\bar{a})}{w(\bar{b}) + w(\bar{a})}$$, where $w(\bar{b}) + w(\bar{a}) \neq 0$

$A_<$ be interpreted as the grade of acceptability of the first interval number to be inferior to the second interval number. Also if $A(\bar{a} \le \bar{b}) \ge 0$, then $\bar{a} \le \bar{b}$ and if $A(\bar{b} \ge \bar{a}) \ge 0$ then $\bar{b} \le \bar{a}$.

Definition:2.3

` Let us define for any two intervals $\bar{a} = [a_L, a_R]$, $\bar{b} = [b_L, b_R]$ in IR and for $* \in \{+,-,.,\div\}$ the arithmetic operations on $\bar{a}$ and $\bar{b}$ defined as follows.

Addition:

$$\tilde{a} + \tilde{b} = < m(\tilde{a}), w(\tilde{a}) > + < m(\tilde{b}), w(\widetilde{b}) > = < m(\tilde{a}) + m(\tilde{b}), \max\{w(\tilde{a}), w(\tilde{b}) >$$

Subtraction:

$$\tilde{a} - \tilde{b} = < m(\tilde{a}), w(\tilde{a}) > - < m(\tilde{b}), w(\widetilde{b}) > = < m(\tilde{a}) - m(\tilde{b}), \max\{w(\tilde{a}), w(\tilde{b}) >$$

Multiplication:

$$\tilde{a} \times \tilde{b} = < m(\tilde{a}), w(\tilde{a}) > \times < m(\tilde{b}), w(\widetilde{b}) > = < m(\tilde{a}) \times m(\tilde{b}), \max\{w(\tilde{a}), w(\tilde{b}) >$$

Division:

$$\tilde{a} \div \tilde{b} = < m(\tilde{a}), w(\tilde{a}) > \div < m(\tilde{b}), w(\widetilde{b}) > = < m(\tilde{a}) \div m(\tilde{b}), \max\{w(\tilde{a}), w(\tilde{b}) >$$

Provided $m(\bar{b}) \neq 0$.

Definition: 2.4

Let $\bar{a} = [a_L, a_R] = \{x \in \Re, a_L \le x \le a_R \ \&\ a_L, a_R \in \Re\}$ be an interval on the real line $\Re$ where a_L, a_R are lower and upper bound of the interval.

Let $A = [a_1, b_1]$ and $B = [a_2, b_2]$ then we have

Addition: $A+B = [a_1 + a_2, b_1 + b_2]$

Subtraction: $A - B = [a_1 - b_2, a_2 - b_1]$

Multiplication : $A*B = [x,y]$ where $x = \min\{a_1a_2, a_1b_2, b_1a_2, b_1b_2\}$ & $y = \max\{a_1a_2, a_1b_2, b_1a_2, b_1b_2\}$

Definition: 2.5

Let A = $[a_1,b_1]$ and B = $[a_2,b_2]$ be two interval numbers. Let $\alpha = \frac{a_1 + a_2}{2}$ & $\beta = \frac{b_1 + b_2}{2}$. If $\alpha \leq \beta$ then A ≤ B , if $\alpha \geq \beta$ then A ≥ B.

III. MODIFIED HUNGARIAN METHOD

Algorithm:

Consider a problem of allocating 'm' jobs to 'n' machines where (m>n)

Step:1 Consider the cost as interval numbers.

Step:2 In each column find the minimum interval cost and subtract the same from the respective column.

Step:3 Find the minimum interval cost in each row and subtract the same from the respective row.

Step:4 Draw a minimum number of lines to cover all the zeros present in the table. If the number of lines is equal to the number of rows then go to step:7 otherwise continue step:5.

Step:5 If the number of lines drawn is not equal to the number of rows then find the smallest uncovered interval cost. Subtract this smallest interval cost element from all the uncovered elements and add this to those entire interval costs which are in the intersection of lines.

Step:6 Repeat steps 4 & 5 until the number of lines is equal to the number of rows.

Step:7 For assigning the job, find the row which has only one zero, circle that zero, and cross out all other zeros in that corresponding column if present.

Step:8 If there is a tie(i.e) if any rows have two or more zeros then assign that value which has a minimum cost in the original problem

Step:9 Continue the steps: 7 & 8 until all the assignments are over.

IV. NUMERICAL EXAMPLES

Example:1

Consider the unbalanced problem given in [7]

	J_1	**J_2**	**J_3**	**J_4**
M_1	[2,16]	[10,22]	[2,14]	[2,14]
M_2	[13,15]	[7,13]	[12,28]	[7,13]
M_3	[12,12]	[9,19]	[4,8]	[9,19]

Solution:

The problem is converted into mid-width interval numbers as <m(a) ,w(a)>

	J_1	**J_2**	**J_3**	**J_4**
M_1	<9,7>	<16,6>	<8,6>	<8,6>
M_2	<14,1>	<10,3>	<20,8>	<10,3>
M_3	<12,0>	<14,5>	<6,2>	<14,5>

Applying the steps we have

	J_1	**J_2**	**J_3**	**J_4**
M_1	<0,7>	<6,6>	<2,6>	<0,6>
M_2	<5,7>	<0,3>	<14,8>	<2,6>
M_3	<3,7>	<4,5>	<0,2>	<6,6>

	J_1	J_2	J_3	J_4
M_1	**<0,7>**	<6,6>	<2,6>	**<0,6>**
M_2	<5,7>	**<0,3>**	<14,8>	<2,6>
M_3	<3,7>	<4,5>	**<0,2>**	<6,6>

The final table value is

	J_1	J_2	J_3	J_4
M_1	**<9,7>**	<16,6>	<8,6>	**<8,6>**
M_2	<14,1>	**<10,3>**	<20,8>	<10,3>
M_3	<12,0>	<14,5>	**<6,2>**	<14,5>

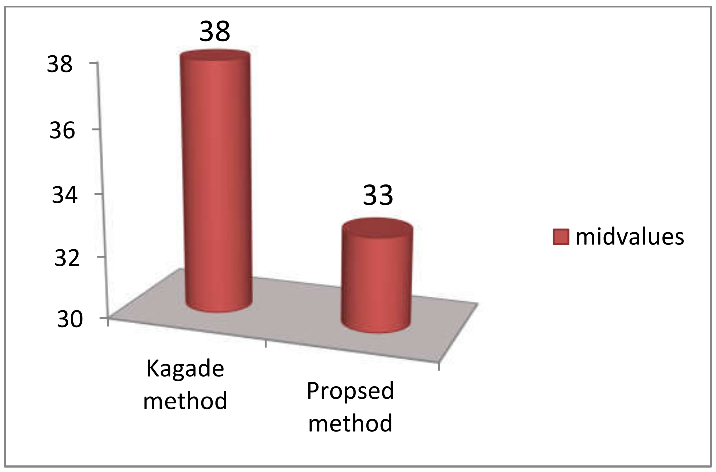
The optimum allocation is $M_1 \rightarrow J_1, J_4$; $M_2 \rightarrow J_2$,$M_3 \rightarrow J_3$

Total cost = <9,7> + <8,6> + <10,3> + <6,2> $\Rightarrow$ <33,18>

Comparison table:

Method	**Kagade [7]**	**Proposed method**
Interval values	[20,56]	[15,51]

It is to be noted that our solution [15, 51] is very much sharper than the solution [20,56] obtained by Kagade [7]



Example: 2

Consider the interval integer unbalanced assignment problem

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
M_1	[299,301]	[249,251]	[179,181]	[319,321]	[269,271]	[189,191]	[219,221]	[259,261]
M_2	[289,291]	[309,311]	[189,191]	[179,181]	[209,211]	[199,201]	[299,301]	[198,191]
M_3	[279,281]	[289,291]	[299,301]	[189,191]	[189,191]	[219,221]	[229,231]	[259,261]
M_4	[289,291]	[299,301]	[189,191]	[239,241]	[249,251]	[189,191]	[179,181]	[209,211]
M_5	[209,211]	[199,201]	[179,181]	[169,171]	[159,161]	[139,141]	[159,161]	[179,181]

Solution:

Step:1 Column minimum

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
M_1	[88,92]	[48,52]	[-2,2]	[148,152]	[108,112]	[48,52]	[58,62]	[78,82]
M_2	[78,82]	[108,112]	[8,12]	[8,12]	[48,52]	[58,62]	[138,142]	[8,12]

M_3	[68,72]	[88,92]	[118,122]	[18,22]	[28,32]	[78,82]	[68,72]	[78,82]
M_4	[78,82]	[98,102]	[8,12]	[68,72]	[88,92]	[48,52]	[18,22]	[28,32]
M_5	[-2,2]	[-2,2]	[-2,2]	[-2,2]	[-2,2]	[-2,2]	[-2,2]	[-2,2]

Step:2 Row minimum

	J_1	**J_2**	**J_3**	**J_4**	**J_5**	**J_6**	**J_7**	**J_8**
M_1	[88,92]	[48,52]	[-2,2]	[148,152]	[108,112]	[48,52]	[58,62]	[78,82]
M_2	[66,74]	[96,104]	[-4,4]	[-4,4]	[36,44]	[46,54]	[126,134]	[-4,4]
M_3	[46,54]	[66,74]	[96,104]	[-4,4]	[6,14]	[56,64]	[64,54]	[56,64]
M_4	[66,74]	[86,94]	[-4,4]	[56,64]	[76,84]	[38,44]	[6,14]	[16,24]
M_5	[-2,2]	[-2,2]	[-2,2]	[-2,2]	[-2,2]	[-2,2]	[-2,2]	[-2,2]

Applying the remaining steps we have

	J_1	**J_2**	**J_3**	**J_4**	**J_5**	**J_6**	**J_7**	**J_8**
M_1	[74,86]	[34,46]	**[-2,2]**	[148,152]	[94,106]	[34,46]	[44,56]	[78,82]
M_2	[52,68]	[82,98]	~~[-4,4]~~	[-4,4]	[22,38]	[32,48]	[112,128]	[-4,4]
M_3	[32,48]	[52,68]	[96,104]	[-4,4]	[-8,8]	[42,58]	[32,48]	[56,64]
M_4	[52,68]	[72,88]	~~[-4,4]~~	[56,64]	[62,78]	[22,38]	**[-8,8]**	[16,24]
M_5	[-2,2]	[-2,2]	[4,16]	[4,16]	[-2,2]	[-2,2]	~~[-2,2]~~	[4,16]

	J_1	**J_2**	**J_3**	**J_4**	**J_5**	**J_6**	**J_7**	**J_8**
M_1	[74,86]	[34,46]	**[-2,2]**	[148,152]	[94,106]	[34,46]	[44,56]	[78,82]
M_2	[52,68]	[82,98]	~~[-4,4]~~	**[-4,4]**	[22,38]	[32,48]	[112,128]	**[-4,4]**
M_3	[32,48]	[52,68]	[96,104]	~~[-4,4]~~	**[-8,8]**	[42,58]	[32,48]	[56,64]
M_4	[52,68]	[72,88]	~~[-4,4]~~	[56,64]	[62,78]	[22,38]	**[-8,8]**	[16,24]
M_5	**[-2,2]**	**[-2,2]**	[4,16]	[4,16]	~~[-2,2]~~	**[-2,2]**	~~[-2,2]~~	[4,16]

The final assignment table is

	J_1	**J_2**	**J_3**	**J_4**	**J_5**	**J_6**	**J_7**	**J_8**
M_1	[299,301]	[249,251]	**[179,181]**	[319,321]	[269,271]	[189,191]	[219,221]	[259,261]
M_2	[289,291]	[309,311]	[189,191]	**[179,181]**	[209,211]	[199,201]	[299,301]	**[198,191]**
M_3	[279,281]	[289,291]	[299,301]	[189,191]	**[189,191]**	[219,221]	[229,231]	[259,261]
M_4	[289,291]	[299,301]	[189,191]	[239,241]	[249,251]	[189,191]	**[179,181]**	[209,211]
M_5	**[209,211]**	**[199,201]**	[179,181]	[169,171]	[159,161]	**[139,141]**	[159,161]	[179,181]

The optimum assignment is

J_3 $\rightarrow M_1$, **J_4** , **J_8**$\rightarrow M_2$, **J_5**$\rightarrow M_3$, **J_1**, **J_7**$\rightarrow M_4$, **J_2**, **J_6**$\rightarrow M_5$.

Job	**J_1**	**J_2**	**J_3**	**J_4**	**J_5**	**J_6**	**J_7**	**J_8**	**Total Cost**
Machine	M_5	M_5	M_1	M_2	M_3	M_5	M_4	M_2	
Cost	[209, 211]	[199, 201]	[179, 181]	[179, 181]	[189, 191]	[139, 141]	[179, 181]	[198, 191]	**[1462, 1478]**

CONCLUSION

In this paper, the modified Hungarian method is applied for solving the unbalanced interval integer assignment problem. This modified approach never assigns the dummy machine in getting the optimum value. Rather all the jobs are assigned to actual machines which can be achieved by allotting multiple jobs to a single machine. Numerical examples are provided to verify the effectiveness of the algorithm.

REFERENCES

[1] Alefeld, G., and Herzberger, J.(1983). Introduction to Interval Computations (Translated by J. Rokne). Academic Press, New York.

[2] Avanish Kumar (2006), A modified method for solving the unbalanced assignment problems, Applied Mathematics and Computation,Vol.176(1), pp.76-82

[3] Betts. N, Vasko.N.J (2016)solving the unbalanced assignment problem:Simple is better , Am.J oper Res:6, 296-299.

[4] Jameer G. Kotwal, Tanuja S. Dhope (2015), Unbalanced assignment problem by using modified approach, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5(7), pp.451-456.

[5] Jayanta Majumdar, Asoke Kumar Bhunia (2012), An alternative approach for unbalanced Assignment problem via genetic algorithm, Applied Mathematics and Computation, Vol. 218(12), pp. 6934-6941.

[6] Kadhirvel.K, Balamurugan,K(2013), Method for Solving Unbalanced Assignment Problems Using Triangular Fuzzy Numbers, Vol. 3(5), pp.359-363.

[7] Kagade. K.L And Bajaj V.H.,(2010), A fuzzy method for solving unbalanced assignment problems with interval valued coefficients, International Journal of Commerce and Business Management,Vol. 3 Issue 1 : 82-87.

[8] Kore, B.G. (2012), A new approach to solve an unbalanced assignment problem, International Journal of Physics and Mathematical Sciences, Vol.2(1), pp. 46-55

[9] Kumar, Avanish (2006). A modified method for solving the unbalanced assignment problems, Applied Mathematics & Computation, 176: 76-82.

[10] *Rabbani et al (2019), Modified Hungarian method for unbalanced Assignment problem with multiple jobs, Applied Mathematics and Computation,493-498.*

[11]Yadaiah, V. and Haragopal, V.V. (2016) , A New Approach of Solving Single Objective Unbalanced Assignment Problem, American Journal of Operations Research, Vol. 6, pp.81-89.