Advances in Mathematics: Scientific Journal **10** (2021), no.2, 1003–1011
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.10.2.29

ASSIGNMENT OF MULTIPLE JOBS SCHEDULING TO A SINGLE MACHINE

Quazzafi Rabbani, Aamir Khan[1], and Abdul Quddoos

ABSTRACT. The existing Modified Hungarian Method for solving the unbalanced assignment problem used multiple jobs assigning process but not used limit of assigned jobs to a single machine, when multiple jobs assign to a single machine then problem occurs that how many jobs can be assigned to a single machine. To overcome this problem, we introduce time parameter on each machine. The present algorithm is implemented on numerical example for better understanding and also coded in Python 3.9.0 programming language.

1. INTRODUCTION

The assignment problem deals with the allocating various resources to various activities on one to one basis i.e. the number of operations are to be assigned to an equal number of operators where each operator performs only one operation. For example suppose a teacher has four students and four tasks. The students differ in efficiency and take different time to perform task. If one task is to be assigned to one student in such a way that the total students hours are minimize, the problem is called assignment problem. The assignment problem

[1]*corresponding author*

2020 *Mathematics Subject Classification.* 65K05, 90C05, 90C08, 90C15, 90C29, 90C70.

Key words and phrases. Unbalanced Assignment Problem, Hungarian assignment method, Modified Hungarian method, Optimization.

Submitted: 09.12.2020; *Accepted:* 24.12.2020; *Published:* 28.02.2021.

is a special case of transportation problem. It is solved using different methods like enumeration method, simplex method, transportation method [3] and Hungarian method [1].

Many authors give different approaches for solving the assignment problem. Among them Hungarian method [1] is more convenient for solving balanced and unbalanced assignment problem. J. Roy and V. Tom [2] give some improvement to the Hungarian method. But while solving unbalanced assignment problem, some jobs are assigned to dummy machine which are not executed at the end of the problem. This looks quite impractical in real life.

Avanish Kumar [4] suggest a method for solving unbalanced assignment problem in which all jobs are executed. In the latest research Quazzafi et al. [5] gives new approach to Hungarian method in which they allow the user to assign more than one job to single machine but they not restrict any machine for doing maximum jobs.

In this paper, particularly we deal with the time management for the jobs which performs by the machine. In this proposed method, we use the formulation of Quazzafi et al. [5] with including time parameter for restrictions on machine. We assign jobs to each machine according to available time of each machine. A step wise algorithm is proposed and solved by taking a numerical example for better understanding.

2. Assumptions

The assumptions of the proposed model are provided below:

- (i) Number of jobs are always greater than the number of machines.
- (ii) Each machine can do more than one job.
- (iii) No same job assign to more than one machine.
- (iv) Each machine at least perform one job and no job is left without assigning to machine.
- (v) A machine can perform a job one after the other if it is to do more than one job.
- (vi) Total assigning time of each machine is less than or equal to its available time.

3. FORMULATION

Given *m* resources and n activities and effectiveness (time) of each resource for each activity, the problem lies in assigning each resource to one activity and if more activities to perform than finish first and after that go for second and so one.

Let t_{ij} be the assigning time of i^{th} machine to j^{th} job where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$; $n > m$, the problems is to find an assignment (which job should be assigned to which machine) so that the total time consumed for performing all tasks is minimum. Here the number of jobs are more than number of machines and one machine can do more than one job and no one job is assign to more than one machine. Each machine has its available time which is used to do the jobs. This can be stated in the form of $m \times n$ cost matrix $[t_{ij}]$ or real numbers, Table 1.

TABLE 1.

	J_1	J_2	$\ldots$	J_n	Time
M_1	t_{11}	t_{12}	$\ldots$	t_{1n}	T_1
M_2	t_{21}	t_{22}	$\ldots$	t_{2n}	T_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
M_m	t_{m1}	t_{m2}	$\ldots$	t_{mn}	T_m

Let x_{ij} denote the i^{th} machine is assigned for j^{th} job such that

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

The Mathematical model is stated as

$$\text{Minimize Z} = \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} x_{ij}; \tag{3.1}$$

Subject to constraints

$$\sum_{j=1}^{n} x_{ij} \geq 1; \text{ for } i = 1, 2, \ldots, \ m; \tag{3.2}$$

(3.3) $$\sum_{i=1}^{m} x_{ij} = 1; \text{ for } j = 1, 2, \ldots, n;$$

(3.4) $$\sum_{j=1}^{n} t_{ij}x_{ij} \leq T_i; \text{ for } i = 1, 2, \ldots,\ m;$$

(3.5) $$x_{ij} = 0 \text{ or } 1.$$

Equation (3.1) shows the objective function, equation (3.2) describes that a machine can do more than one job, equation (3.3) limits that no same job can be allocated to more than one machine and equation (3.4) represents that total assigning time for one machine is always less than or equal to its available time and equation (3.5) shows that it takes only binary number.

4. Algorithm

Step 1- Input m,n

Step 2- Find the minimum time in each row and subtract the same from the respective row it creates at least one zero in each row.

Step 3- Now check each column, if any column is left without creating a zero then select minimum time in that column and subtract from all the elements in that column for obtaining zero.

Step 4- For achieving the ideal matrix; draw the minimum number of lines to cover the zeros.

Step 5- If the numbers of lines are not equal to the number of machines then select the smallest uncovered time and subtract it from each uncovered time and add it at all intersections.

Step 6- Repeat 4 and 5 until the number of lines becomes equal to the number of rows.

Step 7- For assigning the job, find minimum number of zero in each row or column, allocate that zero to the respective machine and subtract the actual time of allocated job to the available time corresponding to that machine.

Step 8- After allocation cross out all the remaining zeros in the corresponding column and if the available time get finished for that machine then cross out all the remaining zeros in that row (total time of allocated jobs to a single machine cannot more than the available time of that machine).

Step 9- In case of tie i.e., any two rows or columns having the same number of zero then assign that zero which has the least time in the original problem. No same job assign to two different machines and no machine left without assigning at least one job.

Step 10- Repeat step 7 to 9 until all the jobs assigned.

Step 11- End of algorithm.

5. Numerical illustration

Here we take the numerical example in which five machines are available for performing eight jobs with associated times (in minutes).

Step 1- Input 5, 8 (Table 2)

TABLE 2.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	Time(min)
M_1	30	25	40	50	35	25	40	25	30
M_2	40	30	20	20	20	45	20	25	40
M_3	20	40	30	40	30	50	30	40	50
M_4	25	20	35	30	25	30	35	30	60
M_5	35	35	50	40	40	60	50	40	80

Step 2- Find the minimum time in each row and subtract the same from the respective row that creates at least one zero in each row.

Step 3- If any column is left without creating a zero then select minimum time in that column and subtract from all the elements in that column for obtaining zero (Table 3)

Step 4- Draw the number of lines to covered all the zeros, here four lines are drawn to cover all zeros and no. of lines are not equal to the no. of machines (Table 4). Now go to step 5

Step 5- Select the smallest cost i.e. 5 from the uncovered costs and subtract from each uncovered cost (Table 5). Now draw lines for covering all zeros. Here five lines are drawn which are equal to number of rows and obtained a required matrix.

TABLE 3.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	Time(min)
M_1	5	0	15	25	10	0	15	0	30
M_2	20	10	0	0	0	25	0	5	40
M_3	0	20	10	20	10	30	10	20	50
M_4	5	0	15	5	5	10	15	10	60
M_5	0	0	15	5	5	25	15	5	80

TABLE 4.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	Time(min)
M_1	5	0	15	25	10	0	15	0	30
M_2	20	10	0	0	0	25	0	5	40
M_3	0	20	10	20	10	30	10	20	50
M_4	5	0	15	5	5	10	15	10	60
M_5	0	0	15	5	5	25	15	5	80

TABLE 5.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	Time(min)
M_1	5	0	15	25	10	0	15	0	30
M_2	20	10	0	0	0	25	0	5	40
M_3	0	20	10	20	10	30	10	20	50
M_4	5	0	15	5	5	10	15	10	60
M_5	0	0	15	5	5	25	15	5	80

Step 6- For assigning the jobs, start with the rows. Find row having only one zero assign that zero and cross the remaining zeros corresponding to that column i.e. J_1 assign to M_3 and subtract the time of assigned job from the available time. Here for machine M_3 available time is 50 minutes and after assigning the job remaining time is 30 minutes and now check the columns having one zero assigned to the respective machine i.e., J_3, J_7 assign to M_2 and J_6 assign to M_1 and subtract time from the available time of corresponding machine (Table 6).

Step 7- In case of tie i.e., any two row or column having the same number of zero, assign that zero which has the least time in the original problem. No same

TABLE 6.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	Time(min)
M_1	5	0	15	25	10	**0**	15	0	5
M_2	20	10	**0**	0	0	25	**0**	5	0
M_3	**0**	20	10	20	10	30	10	20	30
M_4	5	0	10	0	0	5	10	5	60
M_5	0	0	10	0	0	20	10	0	80

job assign to two different machines and no machine left without assigning at least one job.

Step 8- After allocation of the job if the time is not available to do further more jobs for that machine and there is still any location corresponding to that machine than cross it for not assigning more jobs.

Step 9- Continue step 6 to 9 until all jobs are assigned as shown in Table 7.

TABLE 7.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	Time(min)
M_1	5	0	15	25	10	**0**	15	0	5
M_2	20	10	**0**	0	0	25	**0**	5	0
M_3	**0**	20	5	15	5	25	5	15	30
M_4	5	**0**	10	0	**0**	5	10	5	10
M_5	0	0	10	**0**	0	20	10	**0**	0

Step 10- End

The assignment of jobs which minimizes the overall time is shown in Table 8 and bar graph of jobs allocation to the respected machine is shown in Figure 1.

6. CONCLUSION

In this proposed work, a new and simple method was introduced for assigning different activities on the available resources. In this method we include one more parameter as time for each resource and it can be replaced by cost also for assigning cost related problems. This method fulfils the requirements of assigning more activities to one resource within a limit and obtained the optimal result.

TABLE 8.

Machines	Jobs	Time(min)
M_1	j_6	25
M_2	j_3, j_7	20+25
M_3	j_1	20
M_4	j_2, j_5	20+25
M_5	j_4, j_8	40+40
Total Time		**210**

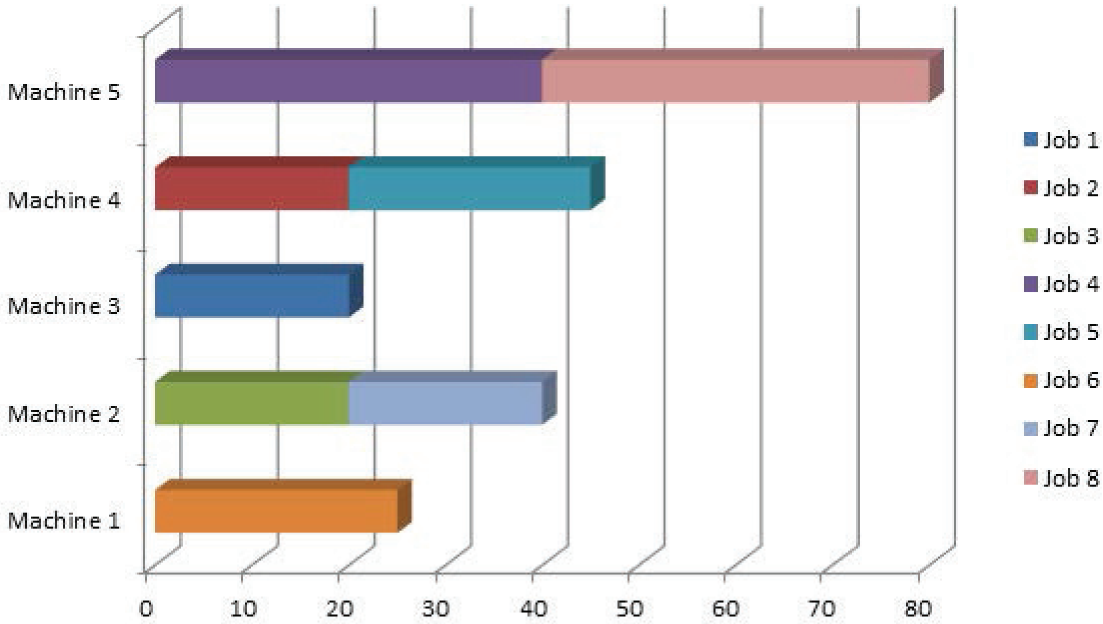

FIGURE 1. Allocations of Jobs to Machines

REFERENCES

[1] H. W. KUHN: *The Hungarian method for the assignment problem*, Nawal Research Logistic Quarterly, **2** (1955), 83-97.

[2] J. ROY, V. TON: *Improving the Hungarian Assignment Algorithm*, Operation Research letters, **5**(4) (1986), 171-175.

[3] J. K. STRAYER: *Transportation and Assignment Problems*, In: Linear Programming and Its Applications. Undergraduate Texts in Mathematics, Springer, New York, NY, 1989.

[4] A. KUMAR: *A modified method for solving the unbalanced assignment problems*, Applied Mathematics and Computation, **176** (2006), 76-82.

[5] Q. RABBANI, A. KHAN, A. QUDDOOS: *Modified Hungarian method for unbalanced assignment problem with multiple jobs*, Applied Mathematics and Computation, **36** (2019), 493-498.

DEPARTMENT OF MATHEMATICS AND STATISTICS
INTEGRAL UNIVERSITY
LUCKNOW, 226026, INDIA
Email address: quazzafi@iul.ac.in

DEPARTMENT OF MATHEMATICS AND STATISTICS
INTEGRAL UNIVERSITY
LUCKNOW, 226026, INDIA
Email address: aamir25dec@gmail.com

DEPARTMENT OF MATHEMATICS AND STATISTICS
INTEGRAL UNIVERSITY
LUCKNOW, 226026, INDIA
Email address: abdulqudoos.or.amu@gmail.com