International Journal Of Core Engineering & Management
Volume-4, Issue-3, June-2017, ISSN No: 2348-9510

TIME MINIMIZATION ASSIGNMENT PROBLEM FOR THREE MACHINES WITH MINMAX OBJECTIVE- A LEXI SEARCH APPROACH

Animoni Nagaraju
Professor, Department of Mathematics and Computer Science
Sree Dattha Group of Institutions, Sheriguda, Ibrahimpatnam ,
Hyderabad, India
animoni_nagaraju@yahoo.co.in

Abstract

Consider A 'time' matrix (t_{ij}) of size m x n gives the time required for job j to be carried out on machine 'i'. Each job has to be done only on one of the machine (i.e. one cannot start processing a job on one machine and halfway shift it to another machine). Also, each machine is required to process not less than m_i^l (at least number of jobs) and not more than m_i^u (at most) jobs; thus, it is permissible that some jobs may have to go unprocessed i.e $\sum_{i=1}^{m} m_i^u \leq n$. With this idea we now formulate the objective for n jobs to be processed on 3 machines i.e m=3, and m<<n, objective function is to minimize the maximum of the total time on the different machines. Here, It should be noted that if $\sum_{i=1}^{m} m_i^u < n$, some of the jobs will necessarily be left unprocessed.

Index Terms: Combinatorial optimization, Time minimizing assignment problem; Bottleneck assignment problem, Generalized assignment Problem, Lexi-Search Approach.

I. INTRODUCTION

The 'usual' assignment problem has the following structure: There are n jobs, and m machines, on any of the machines any of the jobs can be processed. However, the corresponding time/costs are not the same and are given by a cost matrix (C_{ij}) or time matrix (t_{ij}) of order m x n. Each job is to be processed on only one machine. Also, usually each machine is allowed to process not more then n_i jobs i=1:n , the objective is to assign the jobs to the machines in such a way that, subject to the certain constraints, the total cost of the assignment is minimized.

The simplest situation is, when m=n and m_i ,i=1:m, and the well known assignment problem, is solved usually by the Hungarian method, that if m ≠n and m<n with the objective of the total time assignment is minimized is called the time minimization assignment problem(Shalini Arora and Puri-1998).

However, other variations of this problem with changes in the nature of constraints and in the nature of the objective function have been considered from time to time. One such 'generalized assignment problem' is being considered in the present investigation. Since this version has the

International Journal Of Core Engineering & Management
Volume-4, Issue-3, June-2017, ISSN No: 2348-9510

same constraint set of Arora but differs in the objective function.
For each of these problems (i.e.) 3 machines with n jobs is solved by applying lexi-search methodology for TMAP, with minmax objective procedure, in the next section all these procedure are discussed in detail and for this procedure 100 problems are generated randomly and solved by the new developed algorithm, and the optimum solutions are tabulated.
Instead of the costs (C_{ij}), we take the "time required (t_{ij})" However, this also being additive, it is only a change in nomenclature and will no way effect the results.
The classical assignment problem consists of assigning n jobs to an equal number of establishments, one each, so as to meet certain objectives which may be the minimization of the cost incurred or the time taken to complete these jobs. There exist, many practical situations where the ideal conditions do not exist and a certain number of jobs therefore, may be required to be handled by a lesser number of establishments. For major projects, certain well equipped and resourceful establishments may opt to undertake more than one job and in such situations, the decision maker may have to assign the jobs in the best possible manner to meet his objectives. This, and many other similar situations give rise to the variant of the classical assignment problem being discussed in the paper.
TMAP has been considered by many researchers like Nagaraju(2008), Aggarwal [2], Ravindran and Ramaswamy [8] and Bhatia [4] under the usual assumption that work on all the n jobs commence simultaneously. Seshan [9], Shalini Arora [11] considered a generalized version of TMAP when n jobs are considered to be partitioned into p(< n) blocks with precedence constraints on the jobs.
There are, n jobs to be carried out and m=3 (<<n) machines only are available. A 'time' matrix (t_{ij}) of size m x n gives the time required for job j to be carried out on machine 'i'. Each job has to be done only on one of the machine (i.e. one cannot start processing a job on one machine and halfway shift it to another machine). Also each machine is required to process not less than m_i^l (at least number of jobs) and not more than m_i^u (at most) jobs; thus, it is permissible that some jobs may have to go unprocessed i.e $\sum_{i=1}^{m} m_i^u \le n$. With this idea and with the assumed objectives the TMAP is, objective function is to minimize the maximum of the total time on the different machines.

Here, It should be noted that if $\sum_{i=1}^{m} m_i^u < n$, some of the jobs will necessarily be left unprocessed.

II. THEORETICAL DEVELOPMENT

In the this section we demonstrate algorithm with an illustration: Let us consider the TMAP 3 machine 12 jobs situation is shown in Table 1.
PROBLEM: Table - 1: With 3 machines 12 jobs are to be processed

	1	2	3	4	5	6	7	8	9	10	11	12
M1	4	5	3	6	8	5	6	3	9	6	8	10
M2	7	1	1	5	6	3	2	8	7	3	9	2
M3	7	4	3	1	2	4	7	8	11	10	10	9

International Journal Of Core Engineering & Management
Volume-4, Issue-3, June-2017, ISSN No: 2348-9510

We now construct alphabet table, which is an arrangement of times in an increasing order of a TMAP from 1st , 2nd , and 3rd machines. The arrangement of the jobs are done with an index number by not breaking the original sequence of the jobs.

ALPHABET TABLE: Table - 2

S.NO	M1: t_i	M1: j_i	M2: t_i	M2: j_i	M3t_i	M3:j_i
1	3	3	1	2	1	4
2	3	8	1	3	2	5
3	4	1	2	7	3	3
4	5	2	2	12	4	2
5	5	6	3	6	4	6
6	6	4	3	10	7	1
7	6	7	5	4	7	7
8	6	10	6	5	8	8
9	8	5	7	1	9	12
10	8	11	7	9	10	10
11	9	9	8	8	10	11
12	10	12	9	11	11	9

For this problem constraint the requirement is that exactly 3,3 and 3 jobs are to be processed on machines M1,M2 and M3 respectively, an absolute lower bound for any feasible assignment is got as the minimum of maximum machine times of the least 3 ,least 3 and least 3 processing times from three machines, that is the (3+3+4=10) ,(1+1+2=4) and (1+2+3=6), the minimum of maximum of machine times is max(10,4,6)=10. Now the sequence and the job numbers for this objective is {3,8,1}{2,12,9}{4,5,6},the optimal solution is 10

As the timings on the three machines are independent of the sequence of job allotments, bound setting can be done, for each machine separately in parallel or one can first complete assignment on machine 1(say) , then go to machine 2, and then go to machine3 , computing the bound. Compute for an un assignment part on the 1st machine explicitly ,keeping the simpler relating to less efficient bound, calculated once for all, for the M2 , irrespective of job already assigned for the M1,for the M3 , irrespective of jobs already assigned for the M1and M2.

Also, a computationally simpler bound not accumulated the two numbers of jobs yet to be allotted on the machines but, just to multiply the number by the 'next' proceeding time by alphabet table.

In what follows, for the illustration, the allotment is less incomparable made first on (machine) M1, with the constraints minimized bound for M2 and M3, and M1's allotment is completed , and then only , go for more exact bound for the component bound on M2, and then go to M3.

Thus, the search table 'operates' in 3 stages in a search. Let $\{J_1,J_2,J_3,\ldots..,J_9\}$ represents the jobs allotted to the three machines$\{J_1{:}J3\}$ being on M1, {J4:J6} to M2, and $\{J_6{:}J_9\}$ to M3. Naturally each J_i is one of the label jobs with 1:12 and there is (the) no repetition. For instance, J={ 3,8,1,2,7,12,4,5,6} stands for all assigning jobs 3,8,1 to M1 , 2,7,12 to M2 and job 4,5,6, to M3 giving min of max for processing machine times are (3+3+4=10),(1+2+2=5),(1+2+4=7) which is max(10,5,7)=10.

Of course, any number of permutations of these jobs within the same machine, will not change the three total times, also, if the jobs are 're-labeled' separately for each machine, in ascending

International Journal Of Core Engineering & Management
Volume-4, Issue-3, June-2017, ISSN No: 2348-9510

order of processing times, one can avoid the listing with different permutations on the same machine keeping only the first set occurrence of the set in the search table.
This, re-labeling is given in the Alphabet table, are given for M1 the jobs are arranged in increasing order of processing times, and their serial numbers are the 'New" labels for the jobs. Thus, for instance label sequence {3,8,1 | ,2,7,12 | ,4,5,6} Stands for {J_{31},J_{81},J_{11};J_{22},J_{72},J_{12};J_{43},J_{53},J_{63}}.
The search table systematically 'generates' incomplete words using the new labels, for each machine, but records the original job numbers as well accumulate the times included in the word, so far and also bounds for the remaining part of the incomplete word, for the bound for all feasible words in the lexical block, for which the current, incomplete block is a leader. If this bound is greater than a trial solution value on hand, the leader is discarded and the next incomplete word, of the same length, or its next super block leader, as the case may be, is chosen on the current incomplete word. These steps are recorded in the search table presented below, the construction of the search table, as explained above is illustrated below :
The above explanation is illustrated in the following search table:

Search table for 3 machine 12 jobs with minmax objective:

Table-3

J1	J2	J3	J4	J5	J6	J7	J8	J9	
M1	M1	M1	M2	M2	M2	M3	M3	M3	
1	2	3	1	2	3	1	2	3	BOUND
1 3 3 (3) 3+7=10	2 3 8 (6) 6+4=10	3 4 1 (10) 10+0=10	1 1 2 (1) 1+3=4	2 1 3 ®					
				3 2 7 (3) 3+2=5	4 2 12 (5) 5+0=5	1 1 4 (1) 1+5=6	2 2 5 (3) 3+3=6	3 3 3 ®	
								4 4 2 ®	
								5 4 6 (7) 7+0=7	MAX(10,5,7) =10
							3 3 3 ®		
					9 7 1 (10) 10+0=10				BF
					6 3 10 (6) 6+0=6	1 1 4 (1) 1+5=6	2 2 5 (3_) 3 +3=6	3 3 3 ®	
								4 4 2 ®	
								5 4 6 (7)	

International Journal Of Core Engineering & Management
Volume-4, Issue-3, June-2017, ISSN No: 2348-9510

								7+0=7	max(10,6,7)=10 (total fail) BF
							3 3 3 ®		
							5 4 6 (5) 5+7=12		BF
						5 4 6 (4) 4+7=11			BF
					8 6 5 (10) 10+0=10				BF
				7 5 4 (6) 6+6=12					BF
			2 1 3 ®						
			3 2 7 (2) 2+5=7	4 2 12 (4) 4+3=7	5 3 6 (7) 7+0=7	1 1 4 (1) 1+5=6	2 2 5 (3) 3+3=6	3 3 3 ®	
								4 4 3 (7) 7+0=7	TOTAL FAIL(10+7+7)
							3 3 3 ®		
							4 4 2 (5) 5+4=9	5 4 6 ®	
						4 4 2 (4) 4+11=15			BF
					6 3 10 (7) 7+0=7	1 1 4 (1) 1+5=6	2 2 5 (3) 3+3=6	3 3 3 ®	
								5 4 6 (7)	TOTAL FAIL(10+7+7)
								6 7 1 ®	
								7 7 7 (10)	BF
							3 3 3 ®		
							4 4	5 4	

International Journal Of Core Engineering & Management
Volume-4, Issue-3, June-2017, ISSN No: 2348-9510

							2 (5) 5+4=9	6 (9) 9+0=9	TOTAL FAIL(10+7+9)
							5 4 6 (5) 5+7=12		BF
					7 5 4 (10) 10+0=10				BF
				6 3 10 (5) 5+5=10					BF
			4 2 12 (2) 2+6=8	5 3 6 (5) 5+3=8	6 3 10 (8) 8+0=8	1 1 4 (1) 1+5=6	2 2 5 (3) 3+3=6	3 3 3 ®	
								4 4 2 (7) 7+0=7	10+8+7=TF
								7 7 7 (13) 13+0=13	BF
---	---	---	---	---	---	---	---	---	---
2 3 8 (3) 3+9=11									BF
	END								

In this objective, Minimum of the maximum of the M1,M2 and M3 times are considered and lexi-search algorithm is implemented , for the same problem considered above , we illustrate the Objective and the corresponding search table is demonstrated.

In the search table 3 the jobs allotted to the three machines (i.e. M_1, M_2,M3), J_1:J_3 , J_4:J_6 and J_7:J_9 for the first label J_1 allots least time to first index in machine 1 with a time '3' and with the corresponding job number 3 with a preceding time is 3 and the bound is 3+7=10 (i.e. current allotted time +remaining 2 proceeding allotments (J_2:J_3)) for machine 1 .

For the second index label J_2 which allots the next least time of first machine from search table with a time 3 and job number 8, their preceding total time is 6 then the bound is 6+4=10. For third index label J_3 allots the next least time from 1 st machine with a time 4 and job number 1 with a preceding time 10 and the bound is 10+0=10, for machine 2 the maximum time is 10 in this we consider machine 2 for which we have. Now for 4th index label J_4 allots from the next least time of 2nd machine with a time 1 and job number 2 with preceding total time 1 and the corresponding total time for the second machine is 1+3=4, the 5th index label J_5 allots without repetition of job numbers the least index time of 2nd machine with a time 2 and job number 7, for which the preceding time 3, and their proceeding total time is from remaining 1 allotments of 2nd machine is 2.Now, their total time is 3+2=5. For the 6 th index label J_6 allots without repetition of job

numbers in machine 1 and 2, from this the second index will have the next least time for the 2 nd machine with a time 2 and job number 12 whose preceding time is 3 with a proceeding least time having 0 allotments from 2 nd machine, and their total time is 5. for machine 3 the maximum time is max(first machine time=10,second machine time 5)=10 in this way we considered machine 3 for which we have, now, for the 7 th index label J_7 allots without repetition of the job numbers 3,8,1,in machine one,2,7,12 in a machine 2,and the 1st index which have next least time from third machine with the time 1 and job number 4, their preceding time is 1 with a proceeding time of 5 with total time equal to 6. For the 8 th index label J8 allots without repetition of the preceding job numbers 3,8,1,2,7,12,4 from first , second and third machines whose next least time is 2 and job number 5 with a preceding time is 3 with proceeding time as 3 with total time equal to 6. For the 9th index label J_9 allots without repetition of the preceding job numbers 3,8,1,2,7,12,4,5 from first , second and third machines, Now the next least time 4 and job number 6 with a preceding time of 7 ,this is the third machine total time. Now, all the 9 job labels with total of 10+5+7=22. Hence, which is the feasible solution to the problem, Now the bound is max (10, 5, 7) =10. From this the bound falls under J9 label. Since, it is better than the next label proceeding times. So we remove J_9 and J_8 from the word and we allow the next index number without Pre-considering the index 6 of the label J_8, now allow the next index of 7 from J_8, with which we get a new time for the third machine, whose time is lesser then the previous bound, then, which next least time is allowed forJ_9, and then we get a new bound, this bound is better than the previous bound which is considered as a new feasible word and bound, otherwise we remove the previous J_8 label, therefore, J_7 is considered as new label for index 5, and then continuing the new allotments, in this manner from J_9 to J_1 in search of the best word for the best optimal solution.
Now the sequence and the job numbers {3,8,1}{2,7,12}{4,5,6}the optimal solution is max(10,5,7)=10 with total time {3+3+4=10}+{1+2+2=5}+{1+2+4=7}=22.

III. ALGORITHM OF TMAP FOR 3 MACHINES WITH N JOBS:

Step 1: For a three machines with n jobs for the TMAP, we consider a possible number of allowed selections for machine 1 and a number of allowed selections for machine2 and machine 3, out of these three machines, there will be n job selections, and these selections are to be computed.
Step 2: We now construct alphabet table, which is an arrangement of times in a increasing order of the problem for 1st, 2nd and 3 rd machines. After this the arrangement of the jobs is done with an index number by not breaking the original sequence of the jobs.
Step 3: For the various combination of at least and at most constraint given TMAP problems, we obtain the trial solution for the first problem for 3 machines.
Step 4: Applying Lexi-search methodology we follow, Now, Systematically we 'generates' incomplete words, from the search table using the new labels, for each machine, but records original job names as well accumulate the time included in the word so far ,and also bounds for the remaining part of the incomplete word i.e. the bound for all feasible words in the lexical block, for which the current incomplete block is a leader. If this bound is greater than a trial solution value on hand, the leader is discarded and the next incomplete word, of the same length or its next super block leader as the case may be is chosen on the current incomplete word, these steps are recorded in the search table presented below.
Step 5: Using the step 4 we fix objective to obtain a best optimum solution to the minmax criterion i.e. The Objective for three machine case max (T1, T2, T3) is to be minimize; where T1, T2 and T3

International Journal Of Core Engineering & Management
Volume-4, Issue-3, June-2017, ISSN No: 2348-9510

indicate the total time on respective machines, for which obtain the best possible optimum solution.

Step 6 : From the above step, we considered the problem and feasible solution is evaluated for the objective considered in the above and these solutions are compared with other problem feasible solution of step(3), from this we obtain best optimal solution for each objective.

From the above algorithm we obtain the comparison table for above problem

S.NO	**OBJECTIVE**
1	MAX(10,5,7)=10

IV.COMPUTATIONAL EXPERIENCE:

Percentage excess tabulation of objective and its cumulative frequencies verified to 100 problems for objective is shown in the table 5, Percentage excess tabulation of objective and its cumulative frequencies (size 75), verified to 100 problems for the objective

Table 4

OBJECTIVE				
M1	M2	M3	TOT	MIN(MAX)
39	39	39	117	39
19	19	19	57	19
18	17	16	51	18
33	34	35	102	35
21	20	18	59	21
21	21	21	63	21
10	11	11	32	11
26	26	26	78	26
29	28	29	86	29
17	17	17	51	17

CUMULATIVE FREQUENCY

Percentage excess of total and maximum for objective is shown in Table number 5 and clearly observed from figure 1.

International Journal Of Core Engineering & Management
Volume-4, Issue-3, June-2017, ISSN No: 2348-9510

Table5

Ci	Total	Cumtotal	max	Cummax
0.075	11	11	33	33
0.225	18	29	11	44
0.375	32	61	18	62
0.525	34	95	19	81
0.675	5	100	7	88
0.825	0	100	4	92
0.975	0	100	5	97
1.125	0	100	1	98
1.275	0	100	1	99
1.425	0	100	1	100

For this objective, the total and max are shown in the following figure to show how the objectives perform.

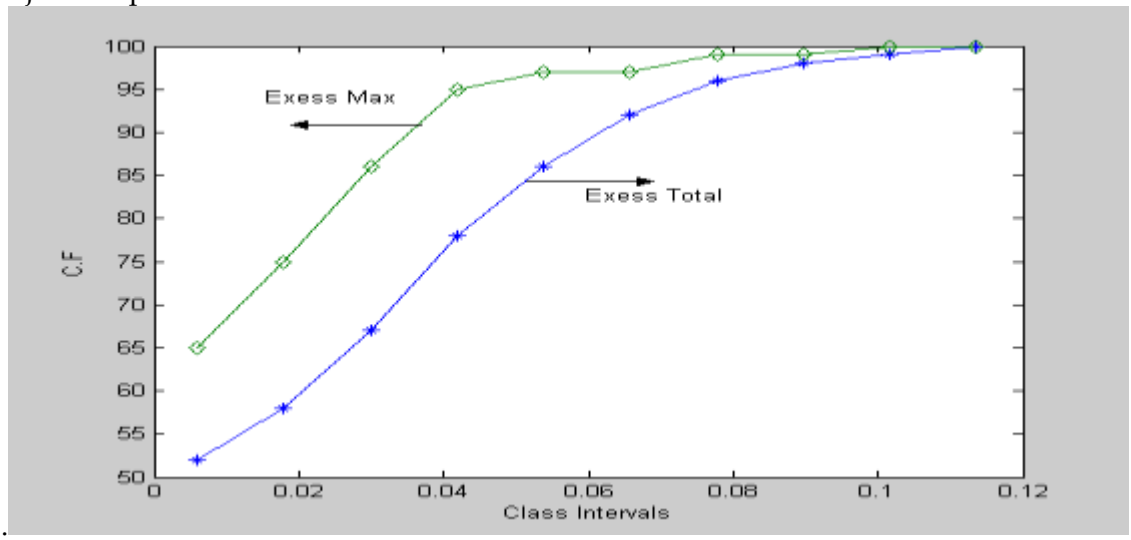

Figure.1 Variation of class intervals versus cumulative frequency

V. CONCLUSIONS

It is observed that time minimization assignment problem with minmax objective with constraints for 3 machines n jobs using the Lexi-search approach for randomly generated problems of various sizes gives a better optimal. We have also, compared the objective of this problem by obtaining the total and minmax time individually computed and found that objective give a better optimum.

REFERENCES

[1] V. Aggarwal, "The assignment problem under categorized jobs", European Journal of Operational Research, 14, 1983, pp193-195.
[2] V. Aggarwal, V.G. Tikekar, L.-F. Hsu, "Bottleneck assignment problems under categorization", Computers and Operations Research 13 (1),1986 , pp 11-26.
[3] O. Berman, D. Einav, G. Handler, "The constrained bottleneck problem in networks", Operations Research 38,1990, pp 178-181.

[4] H.L. Bhatia," Time minimizing assignment problem", Systems and Cybernetics in Management 6,1977, pp.75-83.
[5] S,H. Bokhari, "Assignment problems in distributed and parallel computing", Kluwer Academic Publishers, Boston, 1987.
[6] S.N.N. Pandit, Y.V. Subrahmanyam, "Enumeration of all optimal job sequence", Opsearch 12 (l-2), 1975, pp 35-39.
[7] S.N.N. Pandit, M.S. Murthy, "Allocation of sources and destinations", 8th Annual Convention of Operational Research Society ofIndia, 1975, pp 22-24.
[8] A. Ravindran, V. Ramaswamy, "On the bottleneck assignment problem", Journal of Optimization Theory And Applications 21, 1977, pp 451-458..
[9] C.R. Seshan, "Some generalisations of time minimizing assignment problem", Journal of Operational Research Society 32, 1981, pp 489-494.
[10] Y.V. Subrahmanyam, "Some special cases of assignment problems", Opsearch 16 (1), 1979, pp 45-47.
[11] Shalini Arora, M.C. Puri 1, "A variant of time minimizing assignment problem", European Journal of Operational Research 110, 1998, pp 314-325.
[12] Nagaraju A(2008) " A Lexi search approach to some combinatorial optimization problems" , Unpublished Ph.D thesis, Osmania University, Hyderabad.