ISSN 0976 – 1411

Available online at www.internationaleJournals.com

International eJournals

International eJournal of Mathematics and Engineering 129 (2011) 1169 - 1178

INTERNATIONAL eJOURNAL OF MATHEMATICS AND ENGINEERING

www.internationaleJournals.com

VARIANT OF TIME MINIMIZATION ASSIGNMENT PROBLEM FOR THREE MACHINES DISTANCE MINIMUM -A LEXI SEARCH APPROACH

Nagaraju Animoni *, V. V. Hara Gopal[1] , S. N. Narahari Pandit[2]

*****Department of Mathematics, SV College of computer sciences , Eathabar pally ,Moinabad, Hyderabad,502504, India. E.Mail: animoni_nagaraju@yahoo.co.in

[1] Department of Statistics &CQM, Osmania University, Hyderabad-500007, India,

[2] Center for Quantitative Methods, Osmania University, Hyderabad-500007, India.

Abstract

There are n jobs to be carried out and m (<<n) machines only are available, out of the n jobs k jobs are necessary to be done on the available machines. Consider time matrix (t_{ij}) of size m x n gives the time required for job j is carried out on machine 'i'. Each job has to be done only on one of the machines (i.e. one can not 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) and not more than m_i^u (at most) jobs; thus, it is permissible that some jobs may have to go unprocessed i.e $\sum_{i=1}^{m} m_i^u \le n$. For this approach, we consider the objective is to minimize the distance between accumulate machine times, required for processing the jobs on the machines M_1, M_2 andM_3.

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

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

1. Introduction

In this Paper we discussed variant of time minimization assignment problem with considered new constraint is some jobs are necessary to be processed and rest of them are non necessarily to be processed and Considered objective is distance minimum between accumulate machine times, is discussed using the lexi-search algorithm, this paper explains one of new variation of time minimization assignment problem with considered changes in the nature of constraints and nature of objective function.

The 'usual' assignment problem has the following structure: There are n jobs, and m machines, on any one of the job can be processed on any of the machines.

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 some constraint set is same of Arora(1998) 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 VTMAP, with consider objective, in the next section procedure are discussed in detail and for the procedure 100 problems are generated randomly and solved these by the new developed algorithm, and the optimum solutions are tabulated.

TMAP has been considered by many researchers like Nagaraju(2008), Aggarwal [2], Ravindran and Ramaswamy [7] and Bhatia [3] under the usual assumption that work on all the n jobs commence simultaneously. Seshan [8], Shalini Arora [9] considered a generalized version of TMAP when n jobs are considered to be partitioned into p(< n) blocks with precedence constraints on the jobs.

2. Theoretical development

In the this section we demonstrate algorithm with an illustration: Let us consider the Variant of TMAP for 3 machine 8 jobs situation with constraint jobs 1,3,5 are necessary to be done is shown in Table 1.

Table - 1: With 3 machines 8 jobs are to be processed

PROBLEM

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

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

Table 2

Alphabet table :Necessary jobs

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

Alphabet table : Non Necessary jobs

S.NO	M1: t_i	M1: j_i	M2: t_i	M2: t_i	M3:j_i	M3:j_i
4	3	8	1	2	4	6
5	5	2	2	7	5	8
6	5	6	3	6	6	4
7	6	4	5	4	7	7

8	6	7	8	8	9	2

From the consider constraint we have the requirement is that exactly 7 jobs are processed out of 8 jobs which exactly 2,2,& 3 jobs are to be done on machines M1,M2&M3 respectively, we have possible selections 1,3,5 numbered jobs are necessary to be done out of the given 8 jobs.

Table.3

Selection table for necessary and non necessary **JOBS**

NECESSARY JOBS			NON-NECESSARY JOBS		
M1	M2	M3	M1	M2	M3
0	0	3	2	2	0
0	1	2	2	1	1
0	2	1	2	0	2
1	1	1	1	1	2
1	0	2	1	2	1
2	0	1	0	2	2
1	2	0	1	0	3

From the above table3, there exist 7 sub problems these problem bounds we see in the search table, For this problem the requirement is that exactly 2,2 and 3 jobs are to be processed on machines M1,M2 and M3 respectively, in which necessary and non necessary selections are the least 0 ,least 0 and least 3 processing times on 1^{st} ,2^{nd} 3^{rd} machine and the least 2,least 2 and least 0 processing times on 1^{st} ,2^{nd} 3^{rd} machine which is first sub problem. An absolute lower bound for any feasible assignment is got as the Distance minimum of machine times of the least 2 ,least 2 and least 3 processing times from three machines i.e. the necessary and non necessary selections we compute the feasible solutions, and the best optimal solution job set is :{2,4}{{3},8}{{5,1},6} and the optimal solution is max distance {|11-9|&|9-11|&|11-11|}=2 with total time 11+9+11=31.

As the timings on the three machines are independent of the separate job allotments, bound setting can be done, for each machine separately in parallel or one can first compute assignment on machine 1 (say), then go to machine 2,and machine 3 , computing the bound component for an un assigned part on the 1^{st} machine explicitly ,keeping the simpler and relatively less efficient bound, as calculated once for all, for the M2 , irrespective of jobs already assigned for the M1,similarly for M3, irrespective of jobs already assigned for the M1 and M2.

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

In what follows, in the illustration, allotment is lexicographically made first on (machine) M1, with the constants minimum bound for M2 and M3, and now M1's is allotment is completed , and then only go for more exact bound for the component based on M2 and M3 machines.

Thus, the search table 'operates' in 3 stages, that is, let { J_1,J_2,J3,…..,J5} represents the necessary jobs allotted to the three machines , in which no allotment is being on M1, no allotment is being on M2, and allot {J_1:J_3} to M3,and the non necessary jobs are allotted to the three machines J_4:J_5 being on M1, (J_6:J_7)being on M2, and no allotment being on M3 ,the total jobs represents J_1:J_7. Naturally, each J_i is one of the label jobs with 1:8 and there is to be no repetitions.

From the necessary and non necessary selections we compute the feasible solutions, and the best optimal solution job set is :{2,4}{{3},8}{{5,1},6} and the optimal solution is max {|11-9|&|9-11|&|11-11|}=2 with total time 11+9+11=31.

The search table systematically 'generates' incomplete words using the new labels, for each machine, but records the original job numbers as well accumulates 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 follows as per the lexi search algorithm steps, as explained above is illustrated in search table.

3. Algorithm of TMAP for 3machines with n jobs:

Step 1: For a three machines with n jobs for the VTMAP, we consider a possible number of allowed necessary and non necessary selections for machine 1 , 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 necessary jobs and non necessary jobs for 1st machine, 2nd machine and 3 rd machine. 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 Sub problems of VTMAP problem, we obtain the trial solution for the first sub 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 considered objective. i.e.(objective is to minimize the distance between accumulated machine times, for 2 machines we take the absolute value of $|T_1-T_2|$, which is to be minimized, in the same manner for 3 machines $|T_1-T_2|$,$|T_2-T_3|$&$|T_3-T_1|$ is to be minimized, where as T_1,T_2,T_3 are the Machine times , for consider 3 machine problem procedure is adapted for obtaining the best possible optimum solution.

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

4) The above explanation is illustrated in the following search table:

Search table for 3 machines 8 jobs:

Table 4

J1	J2	J3	J4	J5	J6	J7	
(0,0,3)(2,2,0)							
M1(Non)	M1(Non)	M2(Non)	M2(Non)	M3(ne)	M3(ne)	M3(ne)	Bound
4 3 8 (3) 3+5=8 max(|8-3|,|8-16|,|3-16|)=13	5 5 2 (8) 8+0=8 max(|8-3|,|8-16|,|3-16|)=13	4 1 2 ®					
		5 2 7 (2) 2+3=5 max(|8-5|,|5-16|,|8-16|)=11	6 3 6 (5) 5+0=5 max(|8-5|,|5-16|,|8-16|)=11	1 2 5 (2) 2+14=16 max(|8-5|,|5-16|,|8-16|)=11	2 5 1 (7) 7+9=16 max(|8-5|,|5-16|,|8-16|)=11	3 9 3 (16) 16+0=16 max(|8-5|,|5-16|,|8-16|)=11	11
				cycle fail	cyclefail	cycle fail	
			7 5 4 (7) 7+0=7 max(|8-7|,|7-16|,|8-16|)=9	1 2 5 (2) 2+14=16 max(|8-7|,|7-16|,|8-16|)=9	2 5 1 (7) 7+0=7 max(|8-7|,|7-16|,|8-16|)=9	3 9 3 (16) 16+0=16 max(|8-7|,|7-16|,|8-16|)=9	9
				cycle fail	cycle fail	cycle fail	
			8 8 8 (10) max(|8-10|,|10-16|,|8-16|) 8	1 2 5 (2) 2+14=16 max(|8-10|,|10-16|,|8-16|)=8	2 5 1 (7) 7+0=7 max(|8-10|,|10-16|,|8-16|)=8	3 9 3 (16) 16+0=16 max(|8-10|,|10-16|,|8-16|)=8	8
				cycle fail	cycle fail	cycle fail	
----------	----------	----------	Cycle fail ----------	----------	---------	----------	
5 5 2 (5) 5+5=10 max(|10-13|,|10-16|,|13-16|)=6	6 5 6 (10) 10+0=10 max(|10-13|,|10-16|,|13-16|)=6	4 1 2 ®					
		5 2 7 (2) 2+3=5 max(|10-5|,|5-16|,|16-5|)=11					BF
		6 3 6 ®					
		7 5 4 (5) 5+8=13 max(|10-13|,|10-16|,|13-16|)=6	8 8 8 (13) 13+0=13 max(|10-13|,|10-16|,|13-16|)=6	1 2 5 (2) 2+5+9=16 max(|10-13|,|10-16|,|13-16|)=6	2 5 1 (7) 7+9=16 max(|10-13|,|10-16|,|13-16|)=6	3 9 3 (16) 16+0=16 max(|10-13|,|10-16|,|13-16|)=6	6
			Cycle fail	cycle fail	cycle fail	cycle fail	
		8 8					

		8 (8)					
		cycle fail					
	7 6 4 (11) 11+0=11 max(|11-13|,|11-16|,|13-16|)=5	4 1 2 ®					
		5 2 7 (2) 2+3=5 max(|11-5|,|5-16|,|16-5|)=11					BF
	8 6 7 (11) max(|11-13|,|11-16|,|13-16|)=5	4 1 2 ®					
		5 2 7 ®					
		6 3 6 (3) 3+5=8 max(|11-8|,|8-16|,|16-11|)=8					
		7 5 4 (5) 5+8=13 max(|11-13|,|11-16|,|13-16|)=5	8 8 8 (13) 13+0=13 max(|11-13|,|11-16|,|13-16|)=5	1 2 5 (2) 2+14=16 max(|11-13|,|11-16|,|13-16|)=5	2 5 1 (7) 7+9=13 max(|11-13|,|11-16|,|13-16|)=5	3 9 3 (16) 16+0=16 max(|11-13|,|11-16|,|13-16|)=5	5
	cycle fail	cycle fail	Cycle fail	cycle fail	cycle fail	cycle fail	
6 5 6 (5) 5+6=11 max(|11-13|,13-16|,|11-16|)=5							BF
	cycle fail						
5 5 2 (5) 5+5=10	6 5 6 (10) 10+0=10	1 1 3 (1) 1+1=2	4 1 2 ®				
			5 2 7 (3) 3+0=3 |10-3|=7				BF
			6 3 6 ®				
			7 5 4 (6) 6+0=6 |10-6|=4&|11-6|=5				BF
			8 8 8 (9) |10-9|=1&|9-11|=2	1 2 5 (2) 2+5+4=11	2 5 1 (7) 7+4=11	4 4 6 ®	
						5 5 5 ®	
						6 6	

						4 (13) |9-13|=4	4
						7 7 7 (14) |9-14|=5	BF
	7 6 4 (11)	1 1 3 (1) 1+1=2	4 1 2 ®				
			5 2 7 (3) |11-3|=8				BF
			6 3 6 (4) |11-4|=7				BF
			7 5 4 ® 8 8 8 (9) |11-9|=2&|11-11|=0	1 2 5 (2) 2+5+4=11	2 5 1 (7) 7+4=11	4 4 6 (11) |11-9|=2&|11-11|=0	2
				|11-9|=3		5 5 8 ® 6 6 4 ® 7 7 7 (14) |11-14|=3	BF
					3 9 3 (9) cycle fail		
				2 5 1 (5) 5+9+4=18 |11-18|=7			BF
cycle fail							
(0,2,1)(2,0,2)							
M1(Non)	M1(Non)	M2(ne)	M2(ne)	M3(ne)	M3(Non)	M3(Non)	
4 3 8 (3) 3+5=8	5 5 2 (8) 8+0=8	1 1 3 (1) 1+6=7	2 6 5 (7) 7+0=7 |7-8|=1	1 2 5 ®			
				2 5 1 (5) 5+4+5=14 |14-8|=6			BF
			3 7 1 (8) |8-8|=0	1 2 5 (2) 2+9=11 |8-11|=3			BF
			cycle fail				
		2 6 5 (6) 6+7=13 |8-13|=5					BF
	6 5 6 (8)	1 1 3 (1)	2 6 5 (7)	1 2 5 ®			

	8+0=8	1+6=7	|8-7|=1				
(1,1,1)(1,1,2)							
M1(ne)	M!(Non)	M2(ne)	M2(Non)	M3(ne)	M3(Non)	M3(Non)	
1 3	4 3						
3 (3)	8 (6)						
3+3=6	6+0=6						
	|6-11|=5						BF
	5 5						
	2 (8)						
	8+0=8						
	|8-11|=3						BF
----------	----------	----------	----------	----------	----------	----------	-----
2 4	4 3						
1 (4)	8 (7)						
	|7-11|=4						BF
(1,0,2) (1,2,1)							
M1(ne)	M1(Non)	M2(Non)	M2(Non)	M3(ne)	M3(ne)	M3(Non)	
1 3	4 3						
3 (3)	8 (6)						
	|6-11|=5						BF
	5 5						
	2 (8)						
	|8-11|=3						BF
2 4	4 3						
1 (4)	8 (7)						
	|7-11|=4						BF
	5 5						
	2 (9)						
	|9-11|=2						BF
cycle fail							
(2,0,1)(0,2,2)							
M1(ne)	M1(ne)	M2(Non)	M2(Non)	M3(ne)	M3(Non)	M3(Non)	
1 3	2 4						
3 (3)	1 (7)						
3+4=7	7+0=7						
	|7-11|=4						BF
	3 8	4 1	5 2				
	5 (11)	2 (1)	7 (3)				
	11+0=11	1+2=3	3+0=3				
	|11-11|=0		|11-3|=8				BF
cycle fail							
(1,2,0)(1,0,3)							
M1(ne)	M1(Non)	M2(ne)	M2(ne)	M3(Non)	M3(Non)	M3(Non)	
1 3	4 3						
3 (3)	8 (6)						
	|6-15|=9						BF
	5 5						
	5 (2)						
	|8-15|=7						BF
2 4	4 3						
1 (4)	8 (7)						
	|7-15|=8						BF
		2 5					
		5 ®					

cycle fail	cycle fail	cycle fail					
	END						

In this objective accumulate machine time M1, M2 and M3 are calculated, with that distance minimum of accumulate machine times are found by lexi-search algorithm, which is implemented and the corresponding search table is demonstrated.

In this search table below the necessary jobs and non necessary jobs are allotted to the three machines (i.e. M_1, $M_2$and M3) J_1 : J_2.on M1and J_3:J_4 on M2 and J_5:J_7 on M3. In the first label J_1 does not allots necessary jobs on M1 and M2, since no selections for necessary jobs ,then J_1allots least time to 4th index in machine 1 with non necessary job time as '3' and job number 8 their preceding time is 3 with total allotment time 3+5=8 (i.e. current allotted time from non necessary jobs +remaining non necessary jobs proceeding allotments of machine 1). Similarly, for second index label J_2 we allot next least time of first machine from search table with a time 5 and job number 2, their preceding total time is 8 with total allotment time 8+0=8 .Also, for third index label J_3 allots without repetition of job numbers the least time for the 2 nd machines non necessary job , with a time 2 and job number 7 with the corresponding preceding time 2 and their total allotment time for first machines non necessary jobs is 2+5=5. Now, for next allotment from non necessary jobs, the 4 th index label is J_4 allots next least time of 2nd machines non necessary job with a time 3 and job number 6 whose preceding total time is 5 and Now the total allotment time for 2nd machine is5+0=5, with second machine total time 5 (necessary and non necessary jobs of second machine). Now, the 5 th index label is J_5 which allots without repetition of preceding job numbers with the first least index time of 3rd machines necessary jobs ,with a time 2 and job number 5.Now the total time is 2, the total allotment time for machine 3 is 2+14=16(i.e. current allotted time from necessary jobs +remaining necessary jobs proceeding allotments of machine 3 +remaining non necessary jobs proceeding allotments of machine3). For 6 th index label J_6 allots without repetition of preceding job numbers on machine 1and 2, it is the 2nd index that is next least time of 3rd machine necessary with a time 5 and job number 1 and their preceding total time is 7 with total allotment time is 7+9=16. For the 7 th index label J_7 allots without repetition of preceding job numbers label J_1:J_6,and the 3rd index which is from the next least time of third machine necessary job with the time 9 and job number 3, now their preceding total time is 16, with proceeding time16+0=16 , now all the 7 job labels with a total time 29, which is the feasible solution to the problem, which is known as the word, their bound is max(|8-5|,|5-16|,|16-8|)=11 ,here bound exist with minimization of distance between ($|T_1-T_2|$,$|T_2-T_3|$,$|T_3-T1|$). From the above bound the total word considered for the next allotment is J_7 label, from this next allotments we get better bound, this bound is better than the next label proceeding times, such that , it is removed J_7 and J_6 from the word and allowed next index number, with out consideration of pre-considered J_6, for this the new allowed time is from J_6, from which we get a new bound, this bound is lesser then previous bound, then next least time is allowed forJ_7, and then we get a new bound, this bound is better then the previous bound which will be considered as a new feasible word, otherwise we remove previous J_6 label, and in this J_6 is considered as new label for the next index , and then continuing new allotments in the same manner as discussed above , its search is made from J_7 to J_1, till the best word is attained which will be the best feasible solution. Similarly from other necessary and non necessary selections we compute the feasible solutions, and the best optimal solution job set is :{2,4}{{3},8}{{5,1},6} and the optimal solution is max {|11-9|&|9-11|&|11-11|}=2 with total time 11+9+11=31.

4. Computational Experience: Variant of Time Minimization Assignment Problem for Three Machines Distance Minimum optimum solution table size 70 verified for 100 problems and given below 10 problems optimal solutions

Table -5

OPTIMAL SOLUTION TABLE				
M1	M2	M3	TOT	DIFF
30	31	25	86	6
21	30	31	82	9
6	11	26	43	15
12	14	27	53	13
23	27	11	61	16
24	22	22	68	2
29	11	29	69	18
29	28	22	79	6
27	5	28	60	23
25	20	17	62	5

5. Conclusions:

It is observed that Distance Minimum for Accumulated machine times for the Problem Variant of Time Minimization Assignment Problem for Three Machines and n jobs with the Lexi-search approach for randomly generated problems of various sizes gives a better optimal for consider objective function and constraints.

References

[1] Aggarwal, V. "The assignment problem under categorized jobs", European Journal of Operational Research, 14, 1983, pp193-195.

[2] Aggarwal, V. Tikekar, V.G. Hsu, L.-F. "Bottleneck assignment problems under categorization", Computers and Operations Research 13 (1),1986 , pp 11-26.

[3] Bhatia, H.L." Time minimizing assignment problem", Systems and Cybernetics in Management 6, 1977, pp.75-83.

[4] Nagaraju A(2008) " A Lexi search approach to some combinatorial optimization problems" , Unpublished Ph.D thesis, Osmania University, Hyderabad.

[5] Pandit, S.N.N. Subrahmanyam, Y.V. "Enumeration of all optimal job sequence", Opsearch 12 (l-2), 1975, pp 35-39.

[6] Pandit, S.N.N. Murthy, M.S. "Allocation of sources and destinations", 8th Annual Convention of Operational Research Society of India, 1975, pp 22-24.

[7] Ravindran, A. Ramaswamy, V. "On the bottleneck assignment problem", Journal of Optimization Theory And Applications 21, 1977, pp 451-458..

[8] Seshan, C.R. "Some generalizations of time minimizing assignment problem", Journal of Operational Research Society 32, 1981, pp 489-494.

[9] Shalini Arora, M.C. Puri 1, "A variant of time minimizing assignment problem", European Journal of Operational Research 110, 1998, pp 314-325.