Global Journal of Mathematical Sciences: Theory and Practical.
ISSN 0974-3200 Volume 3, Number 2 (2011), pp. 127-136
© International Research Publication House
http://www.irphouse.com

Variant Constraint of Time Minimization Assignment Problem with Variant Objectives: A Lexi Search Approach

[1*]Animoni Nagaraju, [2]V.V. Hara Gopal and [3]S.N. Narahari Pandit

*[1]Dept. of Mathematics, SV College of Engineering and Technology,
Moinabad, Hyderabad, A.P.-501504, India.
E-mail Id:animoni_nagaraju@yahoo.co.in
[2]Dept. of Statistics, Osmania University, Hyderabad, A.P.-500007, India.
[3]Center for Quantitative Methods, OU, Hyderabad, A.P.-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. 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 $\sum_{i=1}^{m} m_i^u \le n$ .With this idea the problem TMAP developed with different objectives, first one is to minimize the total time required for processing the jobs on the machines, second one 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.

Keywords: Combinatorial optimization; Time minimizing assignment problem; Generalized assignment Problem; Lexi-search method.

Introduction

The time minimization assignment problem is a particular case of assignment problem, The ‘usual’ time minimization assignment problem has the following structure: There are n jobs, and m<n machines, on which any of the jobs can be processed. However, the corresponding times are not the same and are given by a 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 above constraints, the total time of the assignment is minimized (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 same constraint set of Arora and new constraint included k jobs are necessary to done on ‘n’ jobs, but also differ in the objective functions.

For each of these problems 2 machines with n jobs is solved by applying lexi-search methodology for Time minimization assignment problem (TMAP), with two objective procedures. In the next section these two procedures are discussed in detail, for each procedure 100 randomly generated problems are generated and the optimum solutions are tabulated by the lexi search methodology. Also, an illustration of this problem procedure is discussed in detail by listing the search table and alphabet table.

Time minimization assignment problem has been considered by many researchers like 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.

In the next section the defined TMAP is illustrated.

Method (Algorithm of VCTMAP for 2machines with n jobs:)

Step1: For two machines with n jobs where the Jobs are necessary to be done. We consider a possible number of allowed necessary and non necessary selections for machine 1 and a number of allowed necessary and non necessary selections for machine2, out of these two machines, there will be n job selections.

Step2: 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 and 2nd machine. The arrangement of the jobs is done with an index number by not breaking the original sequence of the jobs.

Step3: For these combination of Jobs where all the jobs are necessary to be done, we obtain the trial solution for the first problem and so on , for 2 machine problems.

Step4: Systematically the search table is constructed which ‘generates’ incomplete words, from this table using the new labels, for each machine, records original job names as well, accumulates 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 presented below.

Step5: Using the step 4 we fix only one of the objectives in obtaining a best optimum solution to the specified objective1, objective2 , i.e. In the search table we get the total times for various jobs out of these we pick up the minimum total time which is the objective 1 and this is true for 2 machines. Now the second objective is to minimize the maximum of total time for processing the jobs on machines, where as we consider the 2 machine problem and solve by considering the 2 objectives discussed in the step4 and obtain the optimal solutions for various possible selections. Similar procedure is adopted for objective2 for obtaining the best possible optimum solutions.

Step6: From the above step, we consider the first problem and optimal solution is evaluated for the two objectives considered above and these solutions are compared with other problems.

Lexi-search procedure of VCTMAP

Jobs 1,3,5 are necessary to be done Time minimization assignment problem for 2 machine 12 jobs:

In this section we consider the problem of 2 machines and 12jobs problem with jobs 1,3,5 are necessary to be done TMAP . The solution to this problem is once again by lexi search methodology for considered objectives.

Problem

	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

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

Table1: Alphabet Table: Necessary jobs

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

Table2: Alphabet Table: Non necessary jobs

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

Since the requirement is that exactly 4 ,5 jobs on M1 & M2 are to be processed respectively, we obtain possible selections 1,3,5 jobs are to be done out of the given 12 jobs.

Table3

Necessary Jobs		Non-Necessary Jobs	
M1	M2	M1	M2
0	3	4	2
1	2	3	3
2	1	2	4
3	0	1	5

From the above alphabet table for necessary jobs & Non necessary jobs the least 0 jobs and least 3 jobs processing times in two machines for necessary jobs, similarly for the non necessary jobs the sum of the least 4 (3+5+5+6=19) jobs and least 2 (1+2=3) jobs processing times in two machines, are selected . i.e. the total of the 1st four and 1st five timings on both machines are the sum{0=0}+{1+6+7=14}+ (3+5+5+6=19) + (1+2=3) =36 for first objective 19+17=36 and the second objective is max(19,17)=19.

As the timings on the two 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) and then go to machine 2 , computing the bound component for an un assigned part on the 1st machine explicitly ,keeping the simpler (relatively less efficient bound, is calculated once for all, for the M2 , irrespective of jobs already assigned for the M1).Also, a computationally simpler bound is not accumulated, the number of jobs yet to be already allotted on the machines, but just to multiply this number by the 'next' processing time in alphabet table.

In what follows, in the illustration, allotment is lexicographically made first on machine M1, with the constant minimum bound for M2, and M1 allotment which is completed , and then only go for more exact bound for the components based on M2.

In the search table systematically ’generates’ incomplete words using the new labels, from step 4 process, steps are recorded in the search table presented below as explained above is illustrated below:

Objective 1 search table for 2 machine 12 jobs with condition jobs are necessary to process:

(0,3)(4,2)									
M2(ne)	M2(ne)	M2(ne)	M1(Non)	M1(Non)	M1(Non)	M1(Non)	M2(Non)	M2(Non)	BOUND
1 1	2 6	3 7	4 3	5 5	6 5	7 6	4 1		
3 (1)	5 (7)	1 (14)	8 (3)	2 (8)	6 (13)	4 (19)	2 ®		
1+13=14	7+7=14	14+0=14	14+3+16+3=36	14+8+11+3=36	14+13+6+3=36	14+19+3=36			
							5 2	6 2	
							7 (21)	12 (3)	
							14+21+2=37	14+23=37	37
							6 2		
							12 (21)		
							14+21+3=38		BF
						8 6	4 1		
						7 (19)	2 ®		
						14+19+3=36			
						10 8			
						11 (21)			
						14+21+3=38			BF
					7 6				
					4 (14)				
					14+14+6+3=37				BF
				6 5					
				6 (8)					
				14+8+12+3=37					BF
			5 5						
			2 (5)						
			14+5+17+3=39						BF
(1,2)(3,3)									
M1(ne)	M2(ne)	M2(ne)	M1(Non)	M1(Non)	M1(Non)	M2(Non)	M2(Non)	M2(Non)	
1 3	1 1								
3 (3)	3 ®								
	2 6	3 7	4 3	5 5	6 5	4 1			
	5 (9)	1 (16)	8 (3)	2 (8)	6 (13)	2 ®			
		16+13+5=34	16+3+10+5=34	16+8+5+5=34	16+13+=34				
						5 2	6 2	7 3	
						7 (2)	12 (4)	6 ®	
						16+13+2+5=36	16+13+4+3=36		
								8 3	
								10 (7)	
								16+13+7=36	36
--------	-------	-------	------------	----------	---------	----------	-----------	--------	BF
(2,1)(2,4)									
M1(ne)	M1(ne)	M2(ne)	M1(Non)	M1(Non)	M2(Non)	M2(Non)	M2(Non)	M2(Non)	
1 3	2 4	2 6	4 3	5 5	4 1				
3 (3)	1 (7)	5 (13)	8 (3)	2 (8)	2 ®				
		13+8+8=29	13+3+5+8=29	13+8+8=29					
					5 2	6 2	7 3	8 3	
					7 (2)	12 (4)	6 (7)	10 (10)	31
					13+8+2+8=31	13+8+4+6=31	13+8+7+3=31	13+8+10=31	
				6 5					
				6 (8)					
				13+8+8=29	4 1	5 2	6 2	7 3	
					2 (1)	7 (3)	12 (5)	6 ®	

								8 3	
								10 (8)	
								13+8+8=29	29
----------	--------	--------	-----------	-------------	---------	-------------	---------------	----------	BF
(3,0)(1,5)									
M1(ne)	M1(ne)	M1(ne)	M1(non)	M2(non)	M2(non)	M2(non)	M2(non)	M2(non)	
1 3									
3 (3)									
3+12+8+8=31									BF

In the search table below the necessary jobs allotted to the two machines (i.e. M_1, M_2) are $J_1 : J_3$. the first label J_1 do not allot on machine M1,since no selections for necessary jobs ,then J_1 allots least time to first index in machine 2 for the necessary job with a time is 1 and job number 3 whose preceding time is 1 with bound equal to 1+13+19+3=36 (i.e. current allotted time from necessary jobs +remaining necessary jobs proceeding allotments+ four proceeding allotments for machine 1 with non necessary jobs and other 2 proceeding allotments from machine 2 with non necessary jobs).

Similarly, for second index label J_2 we allot next least time of second machine from search table with a time 6 and job number 5, their preceding total time is 7 with bound 7+7+19+3=36.Also, Similarly applying next index's we obtained at the 7 th index label J_7 allots without repetition of preceding job numbers label J_1:J_6,and the 7th index which is from the next least time of first machine with the time 6 and job number 4, now their preceding total time19+14=33 with proceeding time 3 with a total bound 19+14+3=36. For the 8 th index the label is J_8 which allots without repetition of the preceding job numbers from first and second machine is allowed as the second machine non necessary jobs whose first index least time 2 and job number 7 with a preceding total time of 21+14=35with a proceeding total time 2 with a total bound 37. Similarly for the 9th index label J_9 allots without repetition of the preceding job numbers from first and second machine is allowed and the next least time of second machine non necessary job is 2 and the job number is 12 with a preceding total time 36 with no proceeding time which is taken as 0, now all the 9 job labels with a total bound 36, which is the feasible solution to the problem, which is known as the word, their bound is 36.From the above bound the total word does not consider the next allotment for J_9 label. Since, it is better than next label proceeding times, such that , it is removed J_9 and J_8 from the word and allowed next index number, with out consideration of pre-considered J_8, for this, the new allowed time is from J_8, from which we get a new bound, this bound is lesser then previous bound, then next least time is allowed forJ_9, 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_8 label, and in this J_7 is considered as new label for the next index , and then continuing in this manner the new allotments in the same way as discussed above , its search is made from J_9 to J_1, till the best word is attained which will be the best feasible solution. Similarly from other necessary and non necessary selection feasible solutions, we obtain best optimal solution, that is the jobs are {{3,1},8,6}{{5},2,7,12,10}, for which the optimal solution is{{3+4}+3+5=15}+{6+1+2+2+3=14}=29.

Objective 2 search table for 2 machine 12 jobs with condition jobs are necessary to process:

(0,4)(3,2)										
M1(Non)	M1(Non)	M1(Non)	M1(Non)	M2(ne)	M2(ne)	M2(ne)	M2(Non)	M2(Non)	bound	Total
4 3	5 5	6 5	7 6	1 1	2 6	3 7	4 1			
8 (3)	2 (8)	6 (13)	4 (19)	3 (1)	5 (7)	1 (14)	2 ®			
3+16=19	8+11=19	13+6=19	19+0=19	1+13+3=17	7+7+3=17	14+3=17				
							5 2	6 2		
							7 (16)	12 (18)	max(19,18)=19	37
							16+2=18	18+0=18		
							6 2			
							12 (16)			
							16+3=19		BF	
--------	-----------	----------	---------	------------	----------	-----------	----------	-----------	BF	
5 5										
2 (5)										
5+17=22									BF	
(1,3)(2,3)										
M1(ne)	M1(Non)	M1(Non)	M1(Non)	M2(ne)	M2(ne)	M2(Non)	M2(Non)	M2(Non)		
1 3	4 3	5 5	6 5	1 1						
3 (3)	8 (6)	2 (11)	6 (16)	3 ®						
3+13=16	6+10=16	11+5=16	16+0=16							
				2 6	3 7	4 1				
				5 (6)	1 (13)	2 ®				
				6+7+5=18	13+5=18					
						5 2				
						7 (15)				
						15+5=20			BF	
----------	------------	-----------	----------	---------	----------	-----------	-----------	-----------	BF	
				2 6	3 7	4 1	5 2	6 2		
				5 (6)	1 (13)	2 (14)	7 (16)	12 (18)		
				6+7+5=18	13+5=18	14+4=18	16+2=18	18+0=18	max(17,18)=18	35
							6 2			
							12 (16)			
							16+3=19		BF	
M1(ne)	M1(ne)	M1(Non)	M1(Non)	M2(ne)	M2(Non	M2(Non)	M2(Non)	M2(Non)		
1 3	2 4	4 3	5 5	1 1						
3 (3)	1 (7)	8 (10)	2 (15)	3 ®						
3+4+8=15	7+8=15	10+5=15	15+0=15							
				2 6	4 1					
				5 (6)	2 ®					
				6+8=14						
					5 2	6 2	7 3	8 3		
					7 (8)	12 (10)	6 (13)	10 (16)		
					8+8=16	10+6=16	13+3=16	16+0=16	max(15,16)=16	31
							8 3			
							10 (13)			
							13+5=18		BF	
				cycle fails						
			6 5	1 1						
			6 (15)	3 ®						
			15+0=15							
				2 6	4 1	5 2	6 2	7 3		
				5 (6)	2 (7)	7 (9)	12 (11)	6 ®		
				6+8=14	7+7=14	9+5=14	11+3=14			
								8 3		
								10 (14)		
								14+0=14	max(15,14)=15	29
---------	---------	-----------	--------	---------	--------	--------	---------	--------	BF	
(3,1)(0,5)										
M1(ne)	M1(ne)	M1(ne)	M1(Non)	M2(Non)	M2(Non)	M2(Non)	M2(Non)	M2(Non)		

1 3										
3 (3)										
3+12+3=18									BF	
		END								

As discussed above we consider the objective2 that is Minimum of the maximum of the machine timings of M1 and M2 are considered and the lexi-search algorithm is implemented, for the same problem considered above, we illustrate the respective search table is discussed.

In the search table gives below the necessary jobs and non necessary jobs are allotted to the two machines (i.e. M_1, M_2) are J_1 : J_4.on M1and $J_{5:}J_9$ on M2,In the first label J_1 does not allots necessary jobs on M1,since no selections for necessary jobs ,then J $_1$allots least time to 4th index in machine 1 non necessary jobs with a time ‘3’ and job number 8 their preceding time is 3 with total allotment time as 3+16=19 (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+11=19.Also, for third index label J_3 which allots next least time of 1st machine non necessary job, with a time 5 and job number 6 with corresponding preceding time 13 and their total allotment time for first machine non necessary jobs is 13+6=19. Now, for next allotment from non necessary jobs, the 4th index label J_4 allots least time of 1st machine non necessary job with a time 6 and job number 4 whose preceding total time is 19 and their total allotment time for 1st machine is19+0=19, with first machine total time as 19(necessary and non necessary jobs of first machine), Now the 5th index label is J_5 which allots without repetition of preceding job numbers with the first least index time of 2nd machines necessary jobs ,with a time 1 and job number 3. Now the total time is 1, the total allotment time for machine 2 is 1+(13+3)=17 as discussed in case2 (i.e. current allotted time from necessary jobs +remaining necessary jobs proceeding allotments of machine 2 +remaining non necessary jobs proceeding allotments of machine2). For the 6 th index label J_6 allots without repetition of preceding job numbers on machine 1and 2, it is the 2nd index whose next least time for the 2nd machine whose necessary job with a time 6 and job number 5 with preceding total time as 7 with total allotment time as 7+7+3=17. For the 7 th index label J_7 allots without repetition of preceding job label numbers J1:J6,and the 3rd index which is from the next least time of second machine necessary job with the time 7 and job number 1, now their preceding total time is 14, with proceeding time14+3=17 . For the 8 th index the label is J_8 which allots without repetition of the preceding job numbers from first and second machines, Now , the non necessary jobs are allowed from the second machine with fifth index least time 2 and job number 7 with a preceding total time for the necessary job of second machine time + current time ,it is 16+2=18, with a proceeding total time 2 with a total allotment time for machine 2 is 18. Similarly for the 9th index label J_9 allots without repetition of the preceding job numbers from first and second machine is allowed with the corresponding next least time for second machine non necessary job is 2 and the job number is 12 with a preceding total time 18 with no proceeding time hence it is taken as 0, now all the 9 job labels with a total time 37, which is the feasible solution to the problem, which is known as the word, their bound is max(19,18)=19,here bound exist with maximum of (four allotments in first machine and the five allotments from second machine). From the above bound (19) the total word considered for the next allotment is J9 label, from this next allotment we get a better bound, In this manner we search for a better bound repeatedly and a bound which is better than next label proceeding times, such that , it is removed J_9 and J_8 from the word and allowed next index number, with out consideration of pre-considered labelled J_8, for which the new allowed time is from J_8, from which we get a new bound, this bound value is lesser then previous bound value, then next least time is allowed forJ_9, continuing in this manner 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_8 label, and in this J_7 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_9 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 for the problem is: The jobs are {{3,1},8,6}{{5},2,7,12,10}, for which the optimal solution is {{3+4}+3+5=15}+{6+1+2+2+3=14}=29.

In this observation we found that both the objective cases optimal solutions are same , this is possible for some problems only, the optimal solutions are variants most of the problems for considered objectives, here solution procedure is most important for obtaining the optimal with less time complexity.

Conclusions

It is observed that the Variant Constraint of Time minimization Assignment problem with the Variant objectives for the considered constraints in 2 machines n jobs by Lexi-Search Approach for randomly generated problems of various sizes gives a better optimal than each other objective. We have also, compared the two objectives of this problem by fixing each of the 2 objectives individually and computed the optimum solution of the other and found that in most of the cases objective1 and 2 are seem to give a better optimum. These problems are solved by the lexi-search algorithm in c&c++ code and computed the optimal solutions.

References

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

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

[3] Berman, O. Einav, D. Handler, G., 1990, “The constrained bottleneck problem in networks”, Operations Research 38, pp 178-181.

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