Available online at www.internationaleJournals.com

International eJournals

International eJournal of Mathematics and Engineering 221 (2013) 2170 – 2184

INTERNATIONAL eJOURNAL OF MATHEMATICS AND ENGINEERING

www.InternationaleJournals.com

VARIANT CONSTRAINT OF TIME DEPENDENT TIME MINIMIZATION ASSIGNMENT PROBLEM WITH MINIMIZE OBJECTIVE–A LEXI-SEARCH APPROACH

Animoni Nagaraju*

SV Institute of Engineering and Technology,Eathbarpally,Moinabad,Hyderabad,AP-502504, India.
Email: animoni_nagaraju@yahoo.co.in, svgi.org@gmail.com

Abstract

The variant constraint of time dependent time minimization assignment problem is a 3-dimensional assignment problem, where the time matrix t_{ijk} of size m x n gives the time required for job 'j' is to be carried out on machine 'i ' at particular time 'k'. There are n jobs to be carried out when, only m (<<n) machines are available, with involved new constraint which p<n jobs are necessary to be done and n-p jobs are non necessarily processed on machines.

And Each job has to be done only on one of the machine (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 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 we now formulate the objective is to minimize the total time required for processing the jobs on these machines at particular time.

***Keywords: combinatorial optimization,* Time minimizing assignment problem, time dependent tmap , Lexi-search, Alphabet table, Generalized assignment problem.**

1. Introduction:

The standard assignment problem has been generalized in many directions. In the present paper we will present one generalization on "variant constraint of Time dependent time minimization Assignment problem", which is different from the usual Assignment problem.

Assigning one impartment constraint for the time dependent time minimization assignment problem is the variant of time dependent time minimization assignment problem, this is also usually the 3-dimensional assignment problem. The considered constraint takes various directions of the problem, which is p<n jobs are necessary to be done and rest of them are non necessarily processed on machines.

The 'usual' time minimization assignment problem has the following structure: There are n jobs, and m machines (m<n), on any of machine which any of the jobs can be processed. However, the corresponding time matrix (t_{ij}) of order m x n. Each job is to be done on only one machine. Also, usually each machine is allowed to process not more then n_i jobs i=1:n , here the objective is to assign the jobs to the machines in such a way that, subject to the above defined constraints, the total time of the assignment is to be minimized (Shalini Arora and MC.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, such problem is variant of time dependent time minimization assignment problem.

Instead of the costs (C_{ij}), we take the "time required (t_{ij})" How ever, this also being additive, it is only a change in nomenclature and will no way affect the results.

In this paper we study the following:

" the variant constraint of time dependent time minimization assignment problem is a 3-dimensional assignment problem, where the time matrix(t_{ijk}) of size m x n gives the time required for job 'j' is to be carried out on machine 'i ' at particular time 'k'. There are n jobs to be carried out when, only m (<<n) machines are available. In which p<n jobs are necessary to be done and rest of them are non necessarily processed on machines.

Each job has to be done only on one of the machine (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 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 is to minimize the total time required for processing the jobs on these machines at particular time".

In this above problem we discusses lexi search approach for obtain exact optimum solution to the above mentioned constrained problem. In this we illustrated 2 machine n jobs with various time dependents at particular time cases with new constraint p<n jobs are necessarily to be done and n-p jobs are non necessary to be done , objective is to minimize the total time, in procedure details solutions are verified for 100 randomly generated problems and obtained optimum solutions, that are tabulated by lexi search algorithm.

The scientific contribution of present problem is called the Dynamic User, Optimum Departure time and Route choice (DUO-D&R) assignment. This problem is highly contributed in dynamic transportation and assignment.

In the sequential, We develop a lexi-search algorithm based on the particular time to solve this problem which takes care of the simple combinatorial structure of the problem in Minimum of total time.

2. Numerical Illustration:

The concept and the algorithm developed and illustrated by a numerical example for which , choose k=[1,2,3,4,5] times, the time (cost) array t(i j, k) is given in table.-1

Table – 1: Variant Constraint of Time Dependent time minimization assignment problem

t(i,j,1) =

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

t(i,j,2)=

	1	2	3	4	5	6
M1	3	6	8	5	4	3
M2	1	5	6	3	2	8

t(i ,j,3)=

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

t(i,j,4)=

	1	2	3	4	5	6
M1	5	4	8	5	6	3
M2	2	5	6	3	1	3

t(i,j,5)=

	1	2	3	4	5	6
M1	3	4	9	4	5	2
M2	6	9	4	3	1	5

Let n=5 jobs require processing on m=2 machines, are available with the following processing time(cost) matrix t(i ,j,N) , and consider the constraint. Machine 1 should process at least (m_i^l) 1, but not more than (m_i^u) 2 jobs while , Machine 2 should process at least(m_i^l) 1 but not more than (m_i^u) 3 jobs.

Since the total number of jobs is 6, it follows that one of the 5 jobs has to go unprocessed since at most 2 and at most 3 jobs only can be processed on the two machines.

Let x=(x_{ijk}) be a solution matrix of '1' and '0' ,indicating the allocation of job 'j' to machine i , at time k, x_{ijk}=1 indicating that job j is to be processed on machine 'i' ,

if x_{ijk} =0, then job 'j' should not be processed on machine 'i' at time k.

Obviously, for feasibility, one should have $\sum_{i=1} x_{ijk} \le 1, j = 1,...,n$ and $m_i^l \le \sum_{j=1}^{n} x_{ijk} \le m_i^u$, i=1,2.

and k=1,2,3,4,5….s<= (at most on first machine+ at most on second machine) number of job selections. The corresponding times of processing will be T=T_{ijk} where $T_{ijk}=x_{ijk}*t_{ijk}$ being either 0 or t_{ijk} only.

Now, the Objective is to. min($\sum_i (\sum_k (\sum_j T_{ijk}))$)

Optimal solution (total time)is : 1+1+3+1+3=9.

with job selections for particular times **: 1 (2,1,3) , 1 (5,2,4) ,** 3 **(6,1,2) , 1 (1,2,1) ,3(4,2,5) Which is clearly time (job, machine, particular time)**

For instance when the solution matrix ‘

x(i ,j,1) is $x=\begin{pmatrix}0&0&0&0&0&0\\1&0&0&0&0&0\end{pmatrix}$ we get $T_{ij1}=\begin{pmatrix}0&0&0&0&0&0\\1&0&0&0&0&0\end{pmatrix}$,

x(i ,j,2) is $x=\begin{pmatrix}0&0&0&0&0&1\\0&0&0&0&0&0\end{pmatrix}$ we get $T_{ij2}=\begin{pmatrix}0&0&0&0&0&3\\0&0&0&0&0&0\end{pmatrix}$,

x(i ,j,3) is $x=\begin{pmatrix}0&1&0&0&0&0\\0&0&0&0&0&0\end{pmatrix}$ we get $T_{ij3}=\begin{pmatrix}0&1&0&0&0&0\\0&0&0&0&0&0\end{pmatrix}$,

x(i ,j,4) is $x=\begin{pmatrix}0&0&0&0&0&0\\0&0&0&0&1&0\end{pmatrix}$ we get $T_{ij4}=\begin{pmatrix}0&0&0&0&0&0\\0&0&0&0&1&0\end{pmatrix}$,

x(i ,j,5) is $x=\begin{pmatrix}0&0&0&0&0&0\\0&0&0&1&0&0\end{pmatrix}$ we get $T_{ij4}=\begin{pmatrix}0&0&0&0&0&0\\0&0&0&3&0&0\end{pmatrix}$,

Here jobs 2 and 5 are necessary, other out of five three jobs are non necessary total jobs are processed five.

$\sum T_{1jk}=4$ is total minimum time on machine one and $\sum T_{2jk}=5$ is total minimum time on machine two for k=1,2,3,4,5 particular time

There for $\min(\ (\sum_{k}(\sum_{j}\sum_{j}(T_{1jk},T_{2jk})))=\min(4+5)=9$, whose total is 4+5=9.

This idea is minimum total time on the machines with dependent time.

These applications are most useful in the various industries for allot the sources on the destinations.

3. Algorithm of Variant constraint of Time Dependent TMAP for 2 machines with n jobs:

Step 1: For two machines with n jobs for the Variant constraint of TD TMAP, 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, 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. 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 gives the different

sub problems for VCTD-TMAP problems, we obtain the trial solution for the first problem for 2 machines.

Step 4: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 minimization of total time for VCTDTMAP to obtain a best optimum solution to the objective . i.e. (total time required allows to minimize on the different machines where we get the total time is minimum , this is true for 2 ,3and m machines)

Step 5: Using the step 4 we fix the objective in obtaining a best optimum solution to the specified objective, i.e. In the search table we get the total times for various jobs out of these we pick up the minimize the maximum of the total time on the three machine times which is considered objective and this is true for 3 machines, where as we consider the 3 machine problem and solve by considering the objective discussed in the step4 and obtain the optimal solutions for various possible selections.

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 problems feasible solution of step(3), from this we obtain best optimal solution for considered objective

Now, we implement the above algorithm for a 2 machine and 6 Jobs problem at possible 5 number of selection times to above given problem in the next section.

3.1 Alphabet table:

Table – 2:

ALPHABET TABLE FOR- NECESSARY JOBS				
	M1		M2	
S.NO	Time	(job, machine,ptime)	Time	(job, machine,ptime)
1	1	(2,1,3)	1	(5,2,4)
2	2	(5,1,3)	1	(5,2,5)
3	4	(5,1,2)	2	(5,2,1)
4	4	(2,1,4)	2	(5,2,2)
5	4	(2,1,5)	5	(2,2,2)
6	5	(2,1,1)	5	(2,2,3)
7	5	(5,1,5)	5	(2,2,4)
8	6	(2,1,2)	9	(2,2,1)
9	6	(5,1,4)	9	(5,2,3)
10	8	(5,1,1)	9	(2,2,5)

ALPHABET TABLE FOR - NON NECESSARY JOBS				
	M1		M2	
S.NO	Time	(job, machine,ptime)	Time	(job, machine,ptime)
1	2	(6,1,5)	1	(1,2,2)
2	3	(3,1,1)	1	(4,2,3)
3	3	(1,1,2)	2	(6,2,3)
4	3	(6,1,2)	2	(1,2,4)
5	3	(6,1,4)	3	(4,2,2)
6	3	(1,1,5)	3	(4,2,4)
7	4	(1,1,1)	3	(6,2,4)
8	4	(1,1,3)	3	(4,2,5)
9	4	(4,1,5)	4	(6,2,1)
10	5	(6,1,1)	4	(3,2,5)
11	5	(4,1,2)	5	(1,2,1)
12	5	(4,1,3)	5	(6,2,5)
13	5	(6,1,3)	6	(4,2,1)
14	5	(1,1,4)	6	(3,2,2)
15	5	(4,1,4)	6	(3,2,4)
16	6	(4,1,1)	6	(1,2,5)
17	6	(3,1,3)	7	(3,2,3)
18	8	(3,1,2)	8	(6,2,2)
19	8	(3,1,4)	8	(1,2,3)
20	9	(3,1,5)	9	(3,2,1)

Since the requirement is that exactly 2, & 3 jobs are to be done on machines M1,&M2 respectively, we have possible selections 2,5 jobs are necessary to be done out of the given 6jobs.

Table-3

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

From the above table , an absolute lower bound for any feasible assignment is got As the sum of the least 1 ,least 1 processing times on two machines for necessary jobs and the sum of the least 1,least 2 processing times in two machines for non-necessary jobs. i.e. the total of the Machine1 necessary+Machine2 necessary+Machine1 non necessary+Machine2 non necessary job selection time, that is the sum {(2,1,3)=1}+{(5,2,4)=1}+{(6,1,5)=2}}+{(1,2,2)=1+(4,2,1)=6} =11 is trial solution. Here bound for 4 selections is (1+1+2+1+1=6) and 5 th selection including is 11.

3.2 Search Table:

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, the bounds, 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 as algorithm:

SEARCH TABLE -4

Necessary selections (1,1)		Non- Necessary selections (1 , 2)				
M1	M2	M1	M2	M2	Reason	Total time
Time(job,machine,time)	Time(job,machine,time)	Time(job,machine,time)	Time(job,machine,time)	Time(job,machine,time)		
1 (2,1,3) (1)	1 (5,2,4) (1)	2 (6,1,5) (1)	1-(1,2,2) (1)	1 (4,2,3) (2)		
1+(5)=6 (Bound)	6	6	6		TR	
				2 (6,2,3) (3)	TR	
				2(1,2,4) (4)	TR	
				3 (4,2,2) (5)	TR	
				3 (4,2,4) (6)	TR	
				3 (6,2,4) (7)	TR	
				3 (4,2,5) (8)	TR	
				4 (6,2,1) (9)	JR	
				4 (3,2,5) (10)	TR	
				5 (1,2,1 (11)	JR	
				5 (6,2,5) (12)	JR	
				6 (4,2,1) (13)	BOUND=11	11 (TS)
			1 (4,2,3) (2)		TR	
			2 (6,2,3) (3)		TR	
			2 (1,2,4) (4)		TR	
			3 (4,2,2) (5)			
			10<11 BOUND	3 (4,2,4) (6)	JR	
				3 (6,2,4) (7)	JR	
				3 (4,2,5) (8)	JR	
				4 (6,2,1) (9)	JR	
				4 (3,2,5) (10)	TR	
				5 (1,2,1 (11)	BF	

				12>11 BOUND FAIL		
			3 (4,2,4) (6)		TR	
			3 (6,2,4) (7)		JR	
			3 (4,2,5) (8)		TR	
			4 (6,2,1) (9)		JR	
			4 (3,2,5) (10)		TR	
			5 (1,2,1 (11) 14>11		BF	
		3 (3,1,1) (2)	1-(1,2,2) (1)		TR	
			1 (4,2,3) (2)		TR	
			2 (6,2,3) (3)		TR	
			2(1,2,4) (4)		TR	
			3 (4,2,2) (5)		BF	
			11<11 BOUND FAIL			
		3 (1,1,2) (3)	1-(1,2,2) (1)		JR	
			1 (4,2,3) (2)		TR	
			2 (6,2,3) (3)		TR	
			2(1,2,4) (4)		TR	
			3 (4,2,2) (5)		TR	
			3 (4,2,4) (6)		TR	
			3 (6,2,4) (7)		TR	
			3 (4,2,5) (8)		BF	
			12>11			
		3 (6,1,2) (4)	1 (1,2,1) (1)	1 (4,2,3) (2)	TR	
				2 (6,2,3) (3)	TR	
				2(1,2,4) (4)	TR	
				3 (4,2,2) (5)	TR	
				3 (4,2,4) (6)	TR	
				3 (6,2,4) (7)	TR	

				3 (4,2,5) (8)		
				6+3=9<11	BOUND=9	9 SOL.
			2 (6,2,3) (3)		TR	
			2(1,2,4) (4)		TR	
			3 (4,2,2) (5)		TR	
			3 (4,2,4) (6)		TR	
			3 (6,2,4) (7)		TR	
			3 (4,2,5) (8)		BF	
			12>9			
		3 (6,1,4) (5)			TR	
		3 (1,1,5) (6)			BF	
	1 (5,2,5) (2)	2 (6,1,5) (1)			TR	
	6<9					
		3 (3,1,1) (2)			TR	
	-----	-----	------	----------	-----BF	
	2 (5,2,1) (3)	2 (6,1,5) (1)	-------	--------	BF	
	2 (5,2,2) (4)	2 (6,1,5) (1)	------	------	BF	
	2 (2,2,2) (5)					
	10<9 BOUNDFAIL				BF	
2 (5,1,3) (2)	1 (5,2,4) (1)				JF	
	1 (5,2,5) (2)				JF	
	2 (5,2,1) (3)				JF	
	2 (5,2,2) (4)				JF	
	2 (2,2,2) (5)				BF	
4 (5,1,2) (3)						
9<9 BOUNDFAIL					BF	

Necessary selections (2,0)		Non- Necessary selections (0 , 3)				
M1	M1	M2	M2	M2	Reason	Total time

Time(job,machine,time)	Time(job,machine,time)	Time(job,machine,time)	Time(job,machine,time)	Time(job,machine,time)		
1 (2,1,3) (1)	2 (5,1,3) (2)				TF	
	4 (5,1,2) (3)				BF	
2 (5,1,3) (2)	4 (5,1,2) (3)				TF	
	4 (2,1,4) (4)				BF	
4 (5,1,2) (3)						
12>9					BF	
Necessary selections (0,2)		Non- Necessary selections (2 ,1)				
M2	M2	M1	M1	M2	Reason	Total time
1 (5,2,4)	1 (5,2,5)				JF	
	2 (5,2,1)				JF	
	5 (5,2,2)				JF	
	5 (2,2,2)				BF	
1 (5,2,5)	2 (5,2,1)				JF	
	5 (5,2,2)				JF	
	5 (2,2,2)				BF	
------	-----	-----	-----	----	----	
2 (2,2,2)						
16>9					BF	
	END					

BF= Bound fail, JF= Job repeated (J ob fail), TF= particular time repeated (Time fail)

Optimal solution (total time)is : 1+1+3+1+3=9.

with job selections for particular times : 1 (2,1,3) , 1 (5,2,4) , 3 (6,1,2) , 1 (1,2,1) ,3(4,2,5) Which is clearly time (job, machine, particular time)

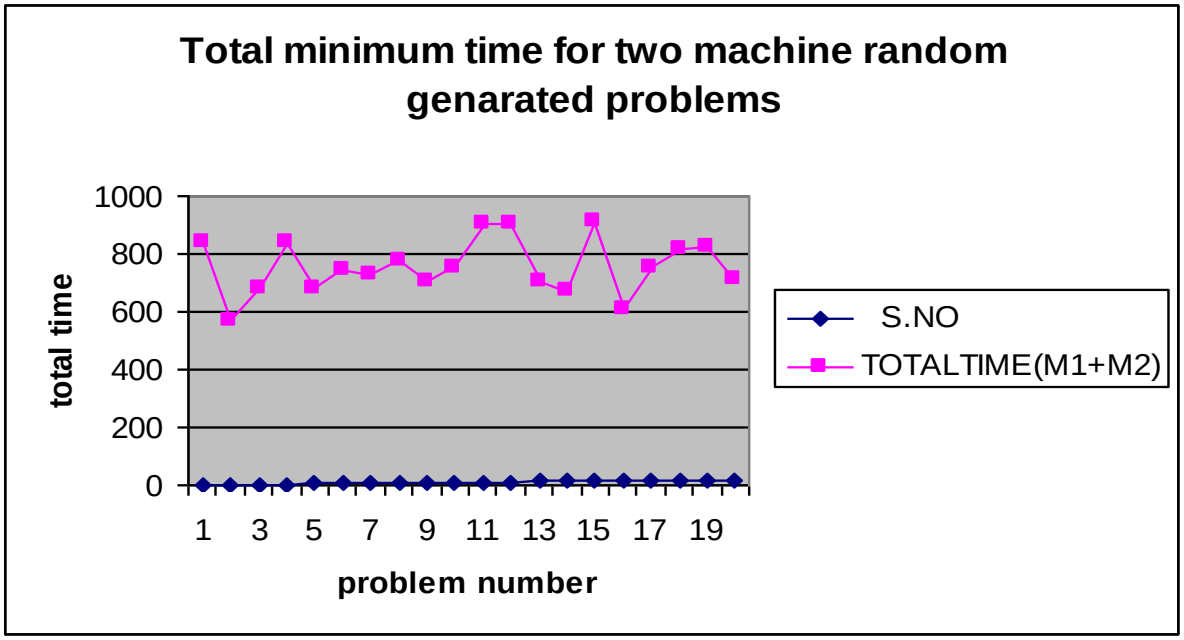
4. The optimum solutions of variant constraint of time dependent time minimization assignment problem of size n=50.

Table 5

S.NO	M1 (Total time)	M2(Total time)	TOTALTIME(M1+M2)
1	419	418	837
2	286	286	572
3	340	341	681
4	420	419	839
5	341	340	681
6	372	371	743
7	365	365	730
8	387	387	774
9	354	354	708
10	376	376	752
11	454	454	908
12	451	450	901
13	351	351	702
14	337	339	676
15	458	456	914
16	305	304	609
17	377	378	755
18	409	409	818
19	412	412	824
20	358	358	716

Optimal Solution Graph

Total minimum time from two machine times is shown in Table and clearly observed from figure 1.



In this figure given optimal line are dependents on the required number of selections of the problem.

5.Conclusions:

With this approach of Variant constraint of Time dependent time minimization assignment problem with minimize objective is optimum for 2 machines with the said constraints and the same is compared with TMAP by lexi-search approach.

6. 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. "Bottleneck assignment problems under categorization", Computers and Operations Research 13 (1),1986, pp 11-26.

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

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

[5] Balinski, M., Gomory. R.. A primal method for the assignment and transportation problems. *Management Sci.*10, 1964,PP 578-593.

[6] Bokhari, S,H. "Assignment problems in distributed and parallel computing", Kluwer Academic Publishers, Boston, 1987.

[7] Cheung, R.K., Powell. W.B." An algorithm for multistage dynamic networks with random arc capacities, with an application to dynamic management", Oper. Res. 44951C963, 1996.

[8] Kuo W-H , Hsu C-J and Yang D-L "A note on unrelated parallel machine scheduling with time-dependent processing times *" Journal of the Operational Research Society* ,2009, advance online publication ; doi: 10.1057/palgrave.jors.2602576

[9] Mosheiov, G "Minimizing total absolute deviation of job completion times: extensions to position-dependent processing times and parallel identical machines",*Journal of the Operational Research Society* 59,2008, 1422–1424. doi:10.1057/palgrave.jors.2602480 Published online 8 August 2007

[10] Nagaraju Animoni,V.V.Haragopal.,S.N.Naraharipandit, "Variant of time minimization assignment problem for three machines distance minimum – A Lexi search approach", International e Journal of mathematics and engineering 129,2011,pp1169-1178.

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

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

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

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

[15] Subrahmanyam, Y.V. "Some special cases of assignment problems", Opsearch 16 (1), 1979, pp 45-47.

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