ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Constrained Three Dimensional Job Assignment Model

P. Madhu Mohan Reddy[*1], C. Suresh Babu[1], V.K. Soma Sekhar Srinivas[1], Sundara Murthy M[1]
[*1] Department of Mathematics, Sri Venkateswara University, Tirupati, Andhra Pradesh, India
mmrphdsv@gmail.com

ABSTRACT

We study the problem called "Constrained Three Dimensional Job Assignment Model (CTDJAM)". For this we took M= ($M_1 \cup M_2 \cup M_3 \cup$, … $\cup M_p$) sets of machines, W= $W_1 \cup W_2 \cup W_3 \cup$, … $\cup W_q$ sets of workers and J= $J_1 \cup J_2 \cup J_3 \cup$, … $\cup J_r$ sets of jobs. Let the number of elements (machines) in each set of machines be m_1, m_2 , , … m_p , the number of elements in each set of workers be w_1, w_2 , w_3 , … w_q and the number of elements in each set of jobs are n_1, n_2 , n_3 , … n_r. The total number of machines, workers, jobs are in all sets are taken as m, w, n respectively. Out of n jobs the number of jobs to be assigned is n_0 (n_0<n). Here third dimension is job. In the feasible assignment all the subsets of machines, workers and jobs should be represented. Our objective is to assign n_0 jobs with minimum total cost/time with the above restrictions.

Keywords: Assignment, Dimension, Constrained.

1. INTROUDCTION

Assignment problem is a special type of linear programming. It is concerned of assigning task to facility on a one to one basis in some optimal way. For e.g., A manager has four persons available for four separate jobs. His job is to assign each person to one and only job in such a way that the time and total cost is minimized.

So many researchers have developed the Assignment Problem.

Purusotham [1] had studied the problem called "Pattern Recognition based Lexi-Search Approach to the Variant Multi-Dimensional Assignment Problem". The Multi Dimensional Assignment Problem is a combinatorial optimization problem that is known to be NP –Hard. On that paper he discusses a problem with four dimensions. N jobs can be executed on N machines, at *k* facilities, using *l* concessions. Every job is to be scheduled on some machine at one of the facilities, using some concession. No two jobs can run on the same machine, at the same facility using the same concession. Furthermore, there is a specified maximum number of jobs that can be run at a given facility, and there is a maximum number of jobs that can avail of a given concession. *C (i, j, k, l)* be the cost of allocating job '*i*' on machine '*j*' at facility '*k*' using the concession '*l*'. This is provided as a 4 dimensional array. The objective is to schedule the jobs in such a way that the constraints are met and the cost is minimized. Sobhan Babu [2] had also studied on the assignment is called "A new Approach for Variant Multi Assignment Problem".

In Hungarian method and Khun et al [4]-1955, Labeling process and the Line covering method are widely used to solve the Assignment Problem. Barr et al [5]-1977 have proposed alternating basis algorithm, while Hung et al [6]-1980 proposed a row algorithm based on Relaxation Method. Bertsekas[7]-1981 produced an algorithm for solving the Classical AP resembling the Hungarian Method in some ways but differs substantially in other aspects. Amore general version of cost minimization Assignment Problem is considered by Geeta et al [8]-1993, where in addition to the (hiring) cost of workers performing the jobs, a supervisory cost is also considered.

The time minimization Assignment Problem (TMAP) is another important class of assignment problem. TMAP has been considered by many researchers like Garfinkel[9] (1971),Ravindran et al[10] (1977), Bhatia[11] (1977), Shalini Arora [12](1997) and Balakrishna[13] (2009) under the usual assumption that work on all the n jobs starts simultaneously.

Vidhyullatha[3] had studied the problem called 'Three Dimensional Group Assignment problem". In this problem there is set of machines and set of jobs are equal in number. Number of persons is the third dimension. Each person is assigned a fixed number of jobs to be performed by him. Each job has to be done on each machine by only one person. The time taken to complete all the jobs is the maximum time taken by a person to complete the jobs assigned to him. In her problem symbolically considered as N={1,2,…,n} set of n jobs, M={set of n machines and P= {1,2,…n} set of p persons($p<n$). t(i,j,k), the time taken by the k^{th} person to

complete i^{th} job on j^{th} machine is given. Out of n jobs each person has to complete the assigned number of jobs which totals to n. A person goes to next job only on the completion of the earlier one. Each job has to be completed individually on separate machine. All the persons start working on the assigned jobs simultaneously. The completion time of all the jobs is the maximum among p persons completion time. The objective is to assign n jobs to p persons with minimum total completion time with the above restrictions and it is a mini max problem.

2. PROBLEM DESCRIPTION

In this paper we study the problem called "Constrained Three Dimensional Job Assignment Model (CTDJAM)". For this we took M machines, W workers and J jobs. Again, M machines are classified into p sets of machines. , W Workers are classified into q sets of workers and J jobs are classified into r sets of jobs. i.e The number of sets in machines are p, The number of sets in workers are q, The number of sets in jobs are r. Let the number of elements in each set of machine are M_1, M_2 , M_3 , … M_p . The number of elements in each set of workers are W_1, W_2 , W_3 , … W_q and the number of elements in each set of jobs are N_1, N_2 , N_3 , … $N_{r.}$ Therefore the total number of machines in all sets taken as m, the total number of workers are taken as w and the total number of jobs are taken as n. Symbolically we can write as the following.

In this CTDJAM, we consider M={1,2,3…p} sets of machines, W={1,2,3…q} sets of workers and J={1,2,3…r} set of jobs. The sets of machines M_1, $M_{2,}$ M3…, Mp are considered such that M= $M_1 \cup M_2 \cup M_3 \cup$, … $\cup M_p$ and $1M_i1=m_i$, 1M 1=m. Here m_{1+} m_{2+} $m_{3+\ldots+}$ m_p =m. Similarly the set of workers W={1,2,3…q} set of workers W_1, W_2 , W_3 , … Wq are considered such that W= $W_1 \cup W_2 \cup W_3 \cup$, … $\cup W_q$ and $1W_i1=w_i$, 1W 1=w. Here w_{1+} w_{2+} $w_{3+\ldots+}$ w_p =w and J={1,2,3…r} set of jobs J_1, J_2 , J_3 , … J_r are considered such that J= $J_1 \cup J_2 \cup J_3 \cup$, … $\cup J_p$ and $1J_i1=n_i$, 1J 1=n. Here n_{1+} n_{2+} $n_{3+\ldots+}$ n_p =n.

Out of n jobs the number of jobs to be completed is n_0 (n_0<n) i.e the total assigned number of jobs should be n_0 <n is truncation. t (i,j,k), the cost taken by the i^{th} machine in M is used by j^{th} worker in W for doing the k^{th} job in J is given. Here third dimension is job. The total assigned number of machines in a particular case of machine (M_s) is less than or equal to that number m_s . Similarly the total assigned number of workers in a particular case of worker (W_s) is

ISSN: 2277-9655
Impact Factor: 1.852

less than or equal to that number w_s and the total assigned number of jobs in a particular case of job (J_s) is less than or equal to that number n_s.

In the feasible assignment schedule all the subset of machines should be represented. i.e. in the solution there is at least one machine from each of the p sets of machines. Similarly in the feasible assignment schedule all the subset of workers should be represented. i.e. in the solution there is at least one worker from each of the q sets of workers and in the feasible assignment schedule all the subset of jobs is should be represented. i.e. in the solution there is at least one job from each of the r sets of jobs. Our objective is to assign n_0 jobs with minimum total cost with the above restrictions.

3. MATHEMATICAL FORMULATION

C (i,j,k) means that cost of the i^{th} machine in M is used by j^{th} worker in W for doing the k^{th} job in J. From n jobs we want to assign n_0 jobs. Here n_0 less than n.

$$Minimize Z(X) = \sum_i \sum_j \sum_k C(i,j,k).x(i,j,k) \quad \text{For } i\in M,\ j\in W,\ k\in J \ \ldots \textbf{(1)}$$

Subjected to constraints

$$\sum_i \sum_j \sum_k x(i,j,k) = n_0 \quad \text{For } i\in M,\ j\in W,\ k\in J \quad \ldots \textbf{(2)}$$

Here $n_0 < n$

$$\sum_i \sum_j \sum_k x(i,j,k) \le m_s, i\in M_s \quad \ldots \textbf{(3)}$$

$$\sum_i \sum_j \sum_k x(i,j,k) \le w_s, j\in W_s \quad \ldots \textbf{(4)}$$

$$\sum_i \sum_j \sum_k x(i,j,k) \le n_s, k\in J_s \quad \ldots \textbf{(5)}$$

If x(i,j,k)= 1,$i\in M_s$ then MI(s)=1 and

ISSN: 2277-9655
Impact Factor: 1.852

$$\sum_{i=1}^{p} MI(i) = p \qquad \text{…(6)}$$

If x (i, j, k) = 1, j$\in$ W_s then WI(s) =1 and

$$\sum_{i=1}^{q} WI(i) = q \qquad \text{... (7)}$$

If x (i, j, k)= 1,i$\in$ J_s then JI(s)=1 and

$$\sum_{i=1}^{r} JI(i) = r \qquad \text{…(8)}$$

x (i, j, k)=0 or 1 i$\in$M, j$\in$W, k$\in$J ... **(9)**

The constraint **(1)** is the objective function which measures the minimum time of completion of all the n_0 jobs under the given restrictions.

The constraint **(2)** describes the restriction that the total number of assigned jobs (n_0) less than or equal to the n.

The constraint **(3)** describes in the job assignment schedule the number of machines assigned from the set M_s should be less than its number m_s.

The constraint **(4)** describes in the job assignment schedule the number of workers assigned from the set W_s should be less than its number w_s.

The constraint **(5)** describes in the job assignment schedule the number of jobs assigned from the set J_s should be less than its number n_s.

The constraint **(6)** describes in the assignment schedule all the subset of machines are involved. i.e. in the solution there is at least one machine from each of the p sets of machines.

The constraint **(7)** describes in the assignment schedule all the subset of workers are involved. i.e. in the solution there is at least one worker from each of the q sets of workers.

The constraint **(8)** describes in the assignment schedule all the subset of jobs are involved. i.e. in the solution there is at least one worker from each of the r sets of jobs.

The constraint **(9)** describes the restriction that the i^{th} machine in M is used by j^{th} worker in W for doing the k^{th} job in J then x (i, j, k) =1 otherwise 0.

4.NUMERICAL ILLUSTRATION

The concept and algorithm developed by illustrated by a numerical example for which taken as the number of machine sets p=5 and in each set having the machines are $M_1=3$, $M_2=2$, $M_3=3$,$M_4=2$,$M_5=3$. Therefore the total number of machines are $m_1+m_2+m_3+m_4+m_5=3+2+3+2+3=13=m$. i.e. the total number of machines=13. Similarly the number of workers sets q=4 and in each set having the workers are $W_1=3$, $W_2=4$, $W_3=3$, $W_4=3$.. Therefore the total number of workers are $w_1+w_2+w_3+w_4=3+4+3+2=12=w$. i.e. the total number of workers=12 and the number of jobs sets r=5 and in each set having the jobs are $J_1=3$, $J_2=4$, $J_3=2$, $J_4=4$, $J_5=3$. Therefore the total number of jobs are $n_1+n_2+n_3+n_4+n_5=3+4+2+4+3=17=n$. i.e. the total number of jobs n=17.

In the following numerical example, C (i, j, k)'s taken as non- negative integers it can be easily seen that this is not a necessary condition. C (i, j, k) means the cost/time that i^{th} machine working on j^{th} worker for k^{th} job. The following table represents the requirement of the cost for do the job with respect to corresponding machine and worker. Then the cost array C (i, j, k) is given below.

TABLE-1

D(i,j,1)=

3	1	2	1
2	3	3	-
4	-	5	2
8	7	-	6
-	9	10	-

D(i,j,2)=

-	6	-	1
4	13	1	5
2	-	8	2
1	3	12	-
13	-	4	12

D(i,j,3)=

1	18	13	-
14	7	14	19
-	2	15	3
6	20	5	-
-	4	5	-

D(i,j,4)=

3	-	4	-
11	-	20	8
7	12	5	-
3	-	21	16
13	14	4	6

D(i,j,5)=

-	1	-	6
4	5	11	3
17	18	15	-
5	17	4	8
7	-	-	2

In **table-1**, D (4, 3, 2) =12 means that the cost of the 4^{th} machine set in M is used by 3^{rd} worker set in W for doing the 2^{nd} job set in J is 12. '-'indicates the there is no assignment of the corresponding machine, worker and job.

5. CONCEPTS AND DEFINITION

5.1 Definition of a pattern

An indicator three-dimensional array which is associated with an assignment is called a 'pattern'. A Pattern is said to be feasible if X is a solution. The pattern represented in the table-2 is a feasible pattern. Now T(X) the value of the pattern X is defined as

$$V(x) = \Sigma\Sigma\, D(i, j, k)\, X(i, j, k)$$

The value V(X) gives the total time of the assignment for the solution represented by X. Thus the value of the feasible pattern gives the total time represented by it. In the algorithm, which is developed in the sequel, a search is made for a feasible pattern with the least value. Each pattern of the solution X is represented by the set of ordered triples [(i,j,k)] for which X(i,j,k)=1, with understanding that the other X(i,j,k)'s are zeros.

5.2 Feasible solution

ISSN: 2277-9655
Impact Factor: 1.852

Consider an ordered pair set{(1,2,1),(3,4,1),(4,3,3),(5,3,2),(4,1,4)(2,2,5),(3,1,2),(2,3,1),(4,3,4),(5,3,4)} represents the pattern given in the **table-2**, which is a feasible solution for the above numerical example.

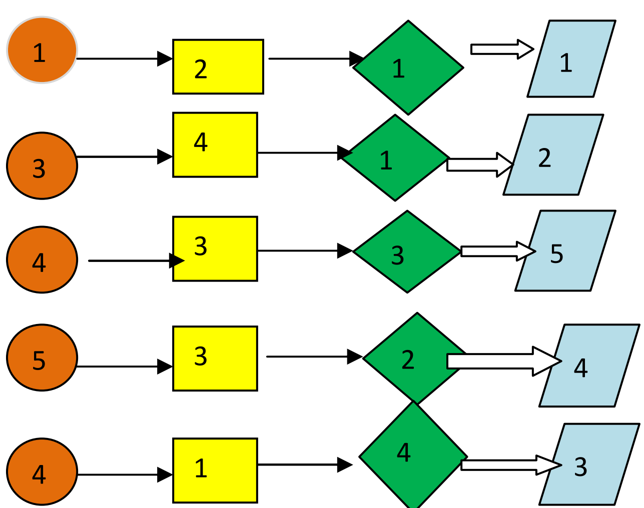
Table-2

$$\mathbf{X\ (i,\ j,1)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{X(i,j,2)}= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X(i,j,3)}= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

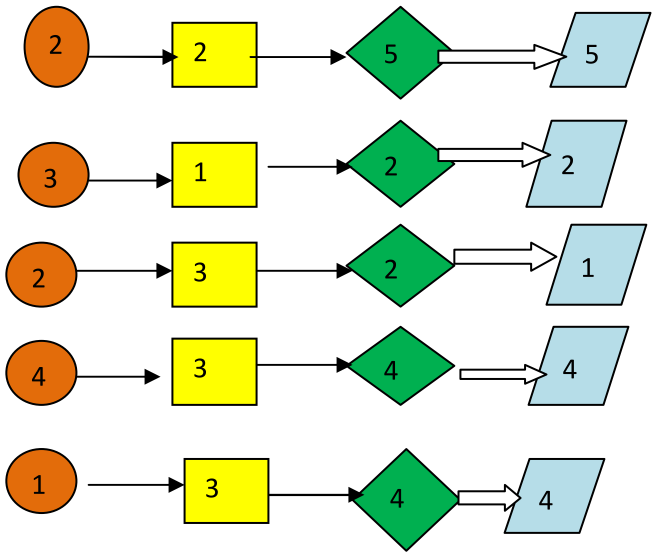
$$\mathbf{X(i,j,4)}= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{X(i,j,5)}= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The following **figure-1** represents a feasible solution. The circle shapes represent machines, rectangle shapes represent workers, diamond shapes represent jobs and parallelogram shape represents the corresponding cost of machine, worker and job. The values in circles indicate name of the machine, values in rectangles indicates name of the worker and values in diamond shapes indicate name of the job.

Figure-1



=



According to the pattern represented in **figure-1** is satisfies all the constraints the section 3. The ordered tripled set represents the cost (1,2,1)=1,(3,4,1)=2,(4,3,3)=5,(5,3,2)=4,(4,1,4)=3,(2,2,5)=5,(3,1,2)=2,(2,3,2)=1,(4,3,4)=4,(5,3,4) =4 . The total cost=1+2+5+4+3+5+2+1+4+4=31.

5.3 Alphabet Table

There are M×W×J ordered triples in the three-dimensional array X. For convenience these are arranged in ascending order of their corresponding costs and are indexed from 1 to M×W×J (Sundara Murthy-1979). Let SN= [1, 2, 3,….., M×W×J] be the set of M×W×J indices. Let C be the corresponding array of costs. If a, b∈ SN and a<b then C (a)≤C(b). Also let the arrays M, W, J be the array of indices of the ordered triples represented by SN and CC be the array of cumulative sum of the elements of C. For convenience same notation M, W, J are used for the corresponding array. The arrays SN, C, CC, M, W, and J for the numerical example are given in the table-3. If p∈ SN then (M(p),W(p),J(p)) is the ordered triple and C(a)=C(M(a),W(a),J(a)) is the value of the ordered triple and CC (a) $= \sum_{i=1}^{a} C(i)$.

TABLE-3

SN	C	CC	M	W	J
1	1	1	1	2	1
2	1	2	1	4	1

3	1	3	2	3	2
4	1	4	1	1	3
5	1	5	1	2	5
6	2	7	1	3	1
7	2	9	2	1	1
8	2	11	3	4	1
9	2	13	3	1	2
10	2	15	3	4	2
11	2	17	3	2	3
12	2	19	5	4	5
13	3	22	1	1	1
14	3	25	2	2	1
15	3	28	2	3	4
16	3	31	4	2	2
17	3	34	3	4	3
18	3	37	1	1	4
19	3	40	4	1	4
20	3	43	2	4	5
21	4	47	3	1	1
22	4	51	2	1	2
23	4	55	5	3	2
24	4	59	5	2	3
25	4	63	1	3	4
26	4	67	5	3	4
27	4	71	2	1	5
28	4	75	4	3	5
29	5	80	3	3	1
30	5	85	2	4	2
31	5	90	4	3	3
32	5	95	5	3	3

33	5	100	3	3	4
34	5	105	2	2	5
35	5	110	4	1	5
36	6	116	4	4	1
37	6	122	1	2	2
38	6	128	4	1	3
39	6	134	5	4	4
40	6	140	1	4	5
41	7	147	4	2	1
42	7	154	1	4	2
43	7	161	2	2	3
44	7	168	3	1	4
45	7	175	5	1	5
46	8	183	4	1	1
47	8	191	3	3	2
48	8	199	2	4	4
49	8	207	4	4	5
50	9	216	5	2	1
51	10	226	5	3	1
52	11	237	4	1	2
53	11	248	2	1	4
54	11	259	2	3	5
55	12	271	4	3	2
56	12	283	5	4	2
57	12	295	3	2	4
58	13	308	2	2	2
59	13	321	5	1	2
60	13	334	1	3	3
61	13	347	5	1	4
62	14	361	2	1	3

ISSN: 2277-9655
Impact Factor: 1.852

63	14	375	2	3	3
64	14	389	5	2	4
65	15	404	3	3	3
66	15	419	3	3	5
67	16	435	4	4	4
68	17	452	3	1	5
69	17	469	4	2	5
70	18	487	1	2	3
71	18	505	3	2	5
72	19	524	2	4	3
73	20	544	4	2	3
74	20	564	2	3	1
75	21	585	4	3	4

Let us consider $5 \in SN$. It represents the ordered triple (M(5),W(5),J(5))=(1,2,5).Then C(5)= 1 and CC(5)=5.

5.4. Definition of an Alphabet - Table and a word

Let SN = (1,2,…) be the set of indices, C be an array of corresponding costs of the ordered triples and CC be the array of cumulative sums of elements in C. Let arrays M, W and J be respectively, the row, column and facility indices of the ordered triples. Let $L_k = \{a_1, a_2, - - -- - , a_k\}$, $a_i \in$ SN be an ordered sequence of k indices from SN. The pattern represented by the ordered triples whose indices are given by L_k is independent of the order of a_i in the sequence. Hence for uniqueness the indices are arranged in the increasing order such that $a_i \leq a_{i+1}$, i = 1, 2, - - - -, k-1. The set SN is defined as the "Alphabet-Table" with alphabetic order as (1, 2, - - - -, M×W×J) and the ordered sequence L_k is defined as a "word" of length k. A word L_k is called a "sensible word". If $a_i < a_{i+1}$, for i =1, 2, - - - -, k-1 and if this condition is called “insensible word". A word L_k is said to be feasible if the corresponding pattern X is feasible and same is with the case of infeasible and partial feasible pattern. A Partial word L_k is said to be feasible if the block of words represented by L_k has at least one feasible word or, equivalently the partial pattern represented by L_k should not have any inconsistency.

Any of the letters in SN can occupy the first place in the partial word L_k. Our interest is only in set of words of length at most n-1, since the words of length greater than n-1 are necessarily infeasible, as any feasible pattern can have only n-1 unit entries in it. If $k < n$, L_k is called a partial word and if $k = n$, it is a full length word or simply a word. A partial word L_k represents, a block of words with L_k as a leader i.e. as its first k letters. A leader is said to be feasible, if the block of word, defined by it has at least one feasible word.

5.5. Value of the word

The value of the (partial) word L_k, V (L_k) is defined recursively as V (L_k) = V (L_{k-1}) + TD (a_k) with V (L_o) = 0 where TD (a_k) is the cost array arranged such that TD (a_k) < TD (a_{k+1}). V (L_k) and V(x) the values of the pattern X will be the same. Since X is the (partial) pattern represented L_k by Sundara Murthy – 1979.

5.6. Lower Bound of A partial word LB (L_k)

A lower bound LB (L_k) for the values of the block of words represented by $L_k = (a_1, a_2, - - - - , a_k)$ can be defined as follows.

$$LB (L_k) = V (L_k) + C(a_{k + j})= V (L_k) + CC (a_k + n - k) - CC (a_k)$$

Consider the partial word L_4 = (3, 6, 9, 17) =V (L_4) = 1+2+2+3= 08

$$LB (L_4) = V (L_4) + DC (a_4 + n - k) - DC (a_4)$$

$$= 08+ DC (17+ 8 - 4) - DC (17) = 08+ CC (21) - CC (17) = 08 + 47 – 34= 21$$

Where CC (a_k) = $\sum_{i=1}^{k} C(a_i)$. It can be seen that LB (L_k) is the value of the complete word, which is obtained by concatenating the first (n-k) letters of SN (a_k) to the partial word L_k

5.7 Feasibility criterion of partial word

A recursive algorithm is developed for checking the feasibility of a partial word. $L_{k+1} = (\alpha_1, \alpha_2 \ldots\ldots,\alpha_k, \alpha_{k+1})$ given that L_k is a feasible partial word. We will introduce some more notations which will be useful in the sequel.

ISSN: 2277-9655
Impact Factor: 1.852

IR be an array where IR (i) =1, i ε M represents that ith machine in M is assigned to jth worker in W for doing the kth job in J is 1, otherwise 0.

IC be an array where IC (i) =1, i ε W represents that ith worker in W is assigned to jth machine in M for doing the kth job in J is 1, otherwise 0.

IK be an array where IK (i) = 1, i ε J represents that ith job in J is assigned to jth machine in M for doing the kth worker in W is 1, otherwise 0.

L be an array where L (i) is the letter in ith position of a partial word

Im be an array where Im (i) = α, i ε RA indicates that the machine is used α times up to the ith position of a partial word and $\alpha \leq m_{RA}$.

Iw be an array where Iw (i) = β, i ε CA indicates that the worker is used β times up to the ith position of a partial word and $\beta \leq m_{CA}$.

Ij be an array where Ij (i) = γ, i ε KA indicates that the job is used γ times up to the ith position of a partial word and $\gamma \leq n_{CA}$.

IMX be an array where IMX (i) = 1, i ε M_s indicates that the M_s Machine is assigned.

IWX be an array where IWX (i) = 1, i ε W_s indicates that the W_s Worker is assigned.

IJX be an array where IJX (i) = 1, i ε J_s indicates that the J_s Worker is assigned.

Then for a given partial word $L_k = (\alpha_1, \alpha_2 \ldots\ldots,\alpha_k)$, the values of the arrays IR, IC, IK, L, Im, Iw, Ij, IMX , IWX and IJX are as follows.

- IR(R (α_i)) = 1, $i=1, 2, \ldots k$, otherwise IR (j)=0
- IC(C (α_i)) = 1, $i=1, 2, \ldots k$, otherwise IC (j)=0
- L (i) = α_i,, $i=1, 2, \ldots k$, otherwise L (j) =0.
- Im (i) = α_i,, $i=1, 2, \ldots k$, otherwise Im (j) =0.
- Iw (i) = α_i,, $i=1, 2, \ldots k$, otherwise Iw (j) =0.
- Ij (i) = α_i,, $i=1, 2, \ldots k$, otherwise Ij (j) =0.
- IMX (i) = 1, $i=1, 2, \ldots k$, otherwise IMX (j)=0
- IWX (i) = 1, $i=1, 2, \ldots k$, otherwise IWX (j)=0
- IJX (i) = 1, $i=1, 2, \ldots k$, otherwise IJX (j)=0

The recursive algorithm for checking the feasibility of a partial word L_k is given as follows: In the algorithm first we equate **IX=0**. At the end If **IX=1** then the partial word is feasible, otherwise it is infeasible. For this algorithm we have **RA=R (α_k), CA=C (α_k) and KA= K (α_k) .**

5.8: ALGORITHMS

Algorithm 1: (Checking the Feasibility)

Step 0: IX=0 ***GOTO 2***

Step 2: MN=WN=JN = 0 ***GOTO 4***

Step 4: IS (IR (RA))+1 $\leq m_{RA}$ ***IF YES GOTO 6***
IF NO GOTO 20

Step 6: IS (IC (CA))+1 $\leq w_{CA}$ ***IF YES GOTO 6***
IF NO GOTO 20

Step 8: IS (IK (KA))+1 $\leq n_{KA}$ ***IF YES GOTO 10***
IF NO GOTO 20

Step 10: IS (MX (RA) = 0 ***IF YES, MNA=MN+1 GOTO 12***

IF NO GOTO 12

Step 12: IS (WX (CA) = 0 ***IF YES, WNB=WN+1 GOTO 14***

IF NO GOTO 14

Step 14: IS (JX (KA) = 0 ***IF YES, JNC=JN+1 GOTO 16***

IF NO GOTO 16

Step 16: Is $N_0 - I \geq (p+q+r) - (MNA+WNB+JNC)$

 ISSN: 2277-9655
Impact Factor: 1.852

IF YES GOTO 18

IF NO GOTO 20

Step 18: IX=1

Step 20: STOP

This recursive algorithm is used in Lexi search algorithm to check the feasibility of a partial word. We start the algorithm with a large value say '∞' as a trial value VT. If the value of a feasible word is known, we can as well start with that value as VT. During the search the value of VT is improved. At the end of the search the current value of VT gives the optimal feasible word. We start the partial word $L_1 = (a_1) = (1)$. A partial word L_k is constructed as $L_k = L_{k-1} * (a_k)$ where * indicates concatenation i.e. chain formation. We will calculate the values of V (L_k) and LB (L_k) simultaneously. Then two cases arise one for branching and the other for continuing the search.

1. **LB (L_k) < VT**. Then we check whether L_k is feasible or not. If it is feasible we proceed to consider a partial word of order (k+1), which represents a sub block of the block of words represented by L_k. If L_k is not feasible then consider the next partial word of order by taking another letter which succeeds a_k in the k^{th} position. If all the words of order 'k' are exhausted then we consider the next partial word of order (k–1).

2. **LB (L_k) ≥ VT.** In this case we reject the partial word L_k. We reject the block of word with L_k as leader as not having optimum feasible solution and also reject all partial words of order 'k' that succeeds L_k.

Now we are in a position to develop a Lexi-Search algorithm to find an optimal feasible word.

Algorithm -2 (Lexi - Search Algorithm (LSA))

Step 1: ***(Initialization)***

 ISSN: 2277-9655
Impact Factor: 1.852

The arrays SN, C, CC, M, W, J, m, w, p, q and r are made available. IR, IC, IK, IMX, IWX, IJX, L, V, N_0 and LB are initialized to zero. The values I=1, J=0, VT= ∞, MAX

Step 2: *J=J+1*
IS (J>MAX) *IF YES GOTO 11*
IF NO GOTO 3

Step 3: *L (I) =J*
IS (I=1) *IF YES V (I) =C (J) GOTO 3B*
IF NO GOTO 3A

Step 3A: V (I) =V (I –1) + C (J) *GOTO 3B*

Step 3B: LB (I) =V (I) +CC (J+N_0–I)–CC (J) *GOTO4*

Step 4: *IS (LB (I) ≥VT)* *IF YES GOTO 11*
IF NO GOTO 5

Step 5: *RA=R (J)*
CA=C (J)
KA=K (J) *GOTO 6*

Step 6: *Check the feasibility of L (using algorithm 1)*

IS (IX=0) *IF YES GOTO 2*
IF NO GOTO 7

Step 7: *IS (IX=1)* *IF YES GOTO 8*
IF NO GOTO 2

Step 8: *IS (I=N_0)* *IF YES GOTO 9*
IF NO GOTO 10

Step 9: *L (I) =J*

 ISSN: 2277-9655
Impact Factor: 1.852

L (I) is full length word and is feasible

VT=V (I), record L (I), VT.   GOTO 13

Step 10:   ***IR (RA)=1***

IC (CA)=1

IK (KA)=1

MN= MNA

WN=WNB

JN=JNC

I=I+1   ***GOTO2***

Step 11:   ***IS (I=1)***   ***IF YES GOTO 14***

IF NO GOTO 12

Step 12:   ***I=I –1***   ***GOTO 13***

Step 13:   ***J=L(I)***

RA=R(J)

CA=C(J)

IK (KA)=1

IR(RA)=0

IC(CA)=0

IR(RA)=IR(RA) –1

IS IR(RA)=0   ***IF NO GOTO 2***

IF YES MN=MN-1   GOTO2

IC(CA)=IC(CA) –1   ***IF NO GOTO 2***

IF YES WN=WN-1   GOTO2

IK(KA)=IK(KA) –1   ***IF NO GOTO 2***

IF YES JN=JN-1   GOTO2

Step 14: STOP & END

The current value of VT at the end of the search is the value of the optimal word. At the end if VT= ∞, it indicates that there is no feasible assignment.

5.9 SEARCH TABLE

The working details of getting an optimal word using the above algorithm for the illustrative numerical example is given in the **Table-4** The columns named (1), (2), (3),…, gives the letters in the first, second, third and so on places respectively. The columns R, C give the row, column indices of the letter. The last column gives the remarks regarding the acceptability of the partial words. In the following table A indicates ACCEPT and R indicates REJECT.

Table-4 (Search Table)

SN	1	2	3	4	5	6	7	8	V	LB	R	C	K	REMARK
1	1								1	11	1	2	1	A
2		2							2	11	1	4	1	A
3			3						3	11	2	3	2	A
4				4					4	11	1	1	3	A
5					5				5	11	1*	2	5	R
6					6				6	12	1*	3	1	R
7					7				6	12	2	1	1	A
8						8			8	12	3	4	1	A
9							9		10	12	3	1	2*	R
10							10		10	12	3	4	2*	R
11							11		10	12	3	2	3*	R
12							12		10	13	5	4	5	A
13								13	13	13	1	1	1*	R
14								14	13	13	2	2	1*	R
15								15	13	13	2*	3	4	R

16								16	13	13	4	2	2*	R
17								17	13	13	3	4	3	R
18								18	13	13	1*	1	4	R
19								19	13	13	4*	1	4	R
20								20	13	13	2	4	5*	R
21								21	14	14	3	1	1*	R
22								22	14	14	2	1	2*	R
23								23	14	14	5	3	2*	R
24								24	14	14	5	2	3*	R
25								25	14	14	1*	3	4	R
26								26	14	14	4	3	4	A, VT=14
27							13		11	14*	1	1	1	R,=VT
28						9			8	12	3	1	2	A
29							10		10	12	3	4	2*	R
30							11		10	12	3	2	3*	R
31							12		10	13	5	4	5	A
32								13	13	13	1	1	1	R
33								14	13	13	2	2	1	R
34								15	13	13	2*	3	4	R
35								16	13	13	4	2	2*	R
36								17	13	13	3	4	3*	R
37								18	13	13	1*	1	4	R
38								19	13	13	4	1	4	A,VT=13
39							13		11	14*	1	1	1	R
40						10			8	12	3	4	2	A
41							11		10	12	3	2	3*	R
42							12		10	13*	5	4	5	R,=VT
43						11			10	13*	3	2	3	R,=VT
44					8				6	12	3	4	1	A
45						9			8	12	3	1	2	A

ISSN: 2277-9655
Impact Factor: 1.852

46							10		10	12	3	4	2*	R
47							11		10	12	3	2	3*	R
48							12		10	13*	5	4	5	R,=VT
49						10			8	12	3	4	2	A
50							11		10	12	3	2	3*	R
51							12		10	13*	5	4	5	R,=VT
52						11			8	13*	3	2	3	R,=VT
53					9				6	12	3	1	2	A
54						10			8	12	3	4	2	A
55							11		10	12	3	2	3*	R
56							12		10	13*	5	4	5	R,=VT
57						11			8	13*	3	2	3	R,=VT
58					10				6	13*	3	4	2	R,=VT
59				5					4	12	1	2	5	A
60					6				6	12	1*	3	1	A
61					7				6	12	2	1	1	A
62						8			8	12	3	4	1*	R
63						9			10	14*	3	1	2	R,>VT
64					8				6	12	3	4	1	A
65						9			8	12	3	1	2	A
66							10		10	12	3	4	2*	R
67							11		10	12	3	2	3*	R
68							12		10	13*	5	4	5	R,=VT
69						10			8	12	3	4	2	A
70							11		10	12	3	2	3	R
71							12		10	13*	5	4	5	R,=VT
72						11			8	13*	3	2	3	R,=VT
73					9				6	12	3	1	2	A
74						10			8	12	3	4	2*	R
75						11			8	13*	3	2	3	R,=VT

ISSN: 2277-9655
Impact Factor: 1.852

76					10				6	13*	3	4	2	R,=VT
77				6					5	13*	1	3	1	R,=VT
78			4						3	12	1	1	3	A
79				5					4	12	1*	2	5	R
80				6					5	13*	1	3	1	R,=VT
81			5						3	13*	1	2	5	R,=VT
82		3							2	12	2	3	2	A
83			4						3	12	1	1	3	A
84				5					4	12	1	2	5	A
85					6				6	12	1*	3	1	R
86					7				6	12	2	1	1	A
87						8			8	12	3	4	1	A
88							9		10	12	3	1	2	A
89								10	12	12	3	4	2*	R
90								11	12	12	3	2	3*	R
91								12	12	12	5	4	5*	R
92								13	13	13*	1	1	1	R
93							10		10	12	3	4	2	A
94								11	12	12	3	2	3*	R
95								12	12	12	5	4	3*	R
96								13	13	13*	1	1	1	R,=VT
97							11		10	12	3	2	3	A
98								12	12	12	5	4	5*	R
99								13	13	13*	1	1	1	R,=VT
100							12		10	13*	5	4	5	R,=VT
101						9			8	12	3	1	2	A
102							10		10	12	3	4	2	A
103								11	12	12	3	2	3*	R
104								12	12	12	5	4	5*	R
105								13	13	13*	1	1	1	R

106							11		10	12	3	2	3	A
107								12	10	12	5	4	5*	R
108								13	13	13*	1	1	1	R,=VT
109							12		10	13*	5	4	5	R,=VT
110						10			8	12	3	4	2	A
111							11		10	12	3*	2	3	R
112							12		10	13*	5	4	5	R,=VT
113						11			8	13*	3	2	3	R,=VT
114					8				6	12	3	4	1	A
115						9			8	12	3	1	2	A
116							10		10	12	3*	4	2	R
117							11		10	12	3*	2	3	R
118							12		10	13*	5	4	5	R,=VT
119						10			8	12	3	4	2	A
120							11		10	12	3*	2	3	R
121							12		10	13*	5	4	5	R,=VT
122						11			8	13*	3	2	3	R,=VT
123					9				6	12	3	1	2	A
124						10			8	12	3	4	2	A
125							11		10	12	3	2	3*	R
126							12		10	13*	5	4	5	R,=VT
127						11			8	13*	3	2	3	R,=VT
128					10				6	13*	3	4	2	R,=VT
129				6					5	13*	1	3	1	R,=VT
130			5						3	13*	1	2	5	R,=VT
131		4							2	13*	1	1	3	R,=VT
132	2								1	12	1	4	1	A
133		3							2	12	2	3	2	A
134			4						3	12	1	1	3	A

ISSN: 2277-9655
Impact Factor: 1.852

135				5					4	12	1	2	5	A
136					6				6	12	1*	3	1	R
137					7				6	12	2	1	1	A
138						8			8	12	3	4	1	A
139							9		10	12	3*	1	2	R
140							10		10	12	3*	4	2	R
141							11		10	12	3*	2	3	R
142							12		10	13*	5	4	5	R,=VT
143						9			8	12	3	1	2	A
144							10		10	12	3*	4	2	R
145							11		10	12	3*	2	3	R
146							12		10	13*	5	4	5	R,=VT
147						10			8	12	3	4	2	A
148							11		10	12	3*	2	3	R
149							12		10	13*	5	4	5	R,=VT
150						11			8	13*	3	2	3	R,=VT
151					8				6	12	3	4	1	A
152						9			8	12	3	1	2	A
153							10		10	12	3*	4	2	R
154							11		10	12	3*	2	3	R
155							12		10	13*	5	4	5	R,=VT
156						10			8	12	3	4*	2	R
157						11			8	13*	3	2	3	R,=VT
158					9				6	12	3	1	2	A
159						10			8	12	3	4	2	A
160							11		10	12	3*	2	3	R
161							12		10	13*	5	4	5	R,=VT
162						11			8	13*	3	2	3	R,=VT

ISSN: 2277-9655
Impact Factor: 1.852

163					10				6	13*	3	4	2	R,=VT
164				6					5	13*	1	3	1	R,=VT
165			5						3	13*	1	2	5	R,=VT
166		4							2	13*	1	1	3	R,=VT
167	3								1	13*	2	3	2	R,=VT

At the end of the search the current value of VT is (01+01+01+01+02+02+02+03) = 13 and it is the value of the feasible word ***L_8* = (1, 2, 3, 4, 7, 9, 12, 19)** it is given in 38^{th} row of the search table – 4 and the corresponding order triples are ***(1, 2, 1), (1, 4, 1), (2, 3, 2) (1, 1, 3) , (2, 1, 1)*,(3,1,2),(5,4,5),(4,1,4)** For this optimal feasible word the arrays IR, IC, IK, Im, Iw, Ij, L, MX,WX and JX are given in **table – 5.**

Table – 5

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
IR	**111**	**11**	**1**	**1**	**1**	**-**	**-**	**-**
IC	**1111**	**1**	**1**	**11**	**-**	**-**	**-**	**-**
IK	**111**	**11**	**1**	**1**	**1**	**-**	**-**	**--**
L	**1**	**2**	**3**	**4**	**7**	**9**	**12**	**19**
Im	**3**	**2**	**1**	**1**	**1**	**-**	**-**	**-**
Iw	**4**	**1**	**1**	**2**	**-**	**-**	**-**	**-**
Ij	**3**	**2**	**1**	**1**	**1**	**-**	**-**	**-**
MX	**1**	**1**	**1**	**1**	**1**	**-**	**-**	**-**
WX	**1**	**1**	**1**	**1**	**-**	**-**	**-**	**-**
JX	**1**	**1**	**1**	**1**	**1**	**-**	**-**	**-**

 ISSN: 2277-9655
Impact Factor: 1.852

For an ordered triple set{(1,2,1),(1,4,1),(2,3,2),(1,1,3),(2,1,1)(3,1,2),(5,4,5),(4,1,4)} represents the pattern given in the **table-6**, which is a optimal solution for the above numerical example.

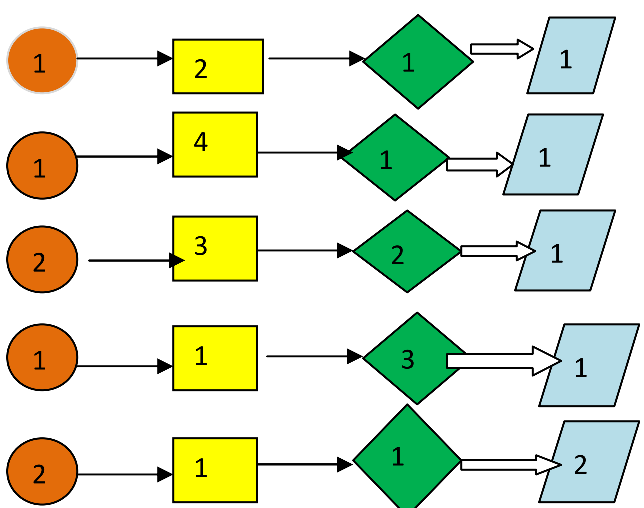
Table-6

$$\mathbf{X\ (i,\ j,1) =} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{X(i,j,2)=} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{X(i,j,3)=} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

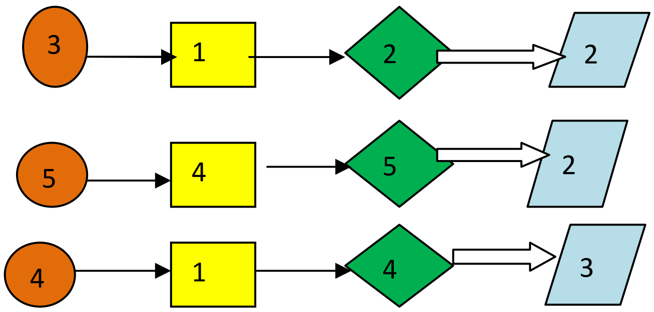
$$\mathbf{X(i,j,4)=} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{X(i,j,5)=} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The following **figure-2** represents a optimal solution. The circle shapes represent machines, rectangle shapes represent workers, diamond shapes represent jobs and parallelogram shape represents the corresponding cost of machine, worker and job. The values in circles indicate name of the machine, values in rectangles indicates name of the worker and values in diamond shapes indicate name of the job.

Figure-2



=



According to the pattern represented in **figure-2** is satisfies all the constraints the section 3. The ordered tripled set represents the cost (1,2,1)=1,(1,4,1)=1,(2,3,2)=1,(1,1,3)=1,(2,1,1)=2,(3,1,2)=2,(5,4,5)=2,(4,1,4)=3. The total cost=1+1+1+1+2+2+2+3=13.

6. CONCLUSION

In this paper, we have studied a model namely "Constrained Three Dimensional Job Assignment Problem ". We have developed a new algorithm which is efficient, accurate and easy to understand. First the model is formulated in to a zero one programming problem. The problem is discussed in detail with help of numerical illustration.

REFERENCES

[1] Purusotham, S et.al. International Journal of Engineering Science and Technology (IJES Vol. 3 No. 8, 2011

[2] Sobhan Babu. K et.al. International Journal on Computer Science and Engineering, (IJCSE) ,Vol. 02, No. 05, 2010, 1633-1640

[3] Vidhyullatha ,A Thesis(Pattern Recognition Lexi-Search Exact Algorithms For Variant ASP and Bulk TP models), Department of mathematics, Sri Venkateswara Uniiversity, Tirupati, 2011.

[4] Kuhn, H.W, the Hungarian Method for the Assignment Problem, Navl Rec. Log.Qtly.2, 83-97, 1955.

[5] Barr, R.S., Glover, F. & Klingman, D, the alternating basis algorithm for assignment Problem, Math. Prog., 13. 1-13, 1977.

[6] Hung, M.S.& Rom,W.O. , Solving the Assignment problem by relaxation. Ops.Res.,18, 969-982, 1980.

[7] Bertsekas, D.P., A New Algorithm for the Assignment problem, Math. Prog.21. 152-171, 1981.

[8] Geetha, S. & Nair, K.P. ,A variation of the Assignment problem, Euro.J. Of Or, 68, 422-426, 1993.

[9] Garfinkel,R.S., an improved algorithm for bottleneck assignment problem, operations research,18, 1717-1751, 1971.

[10] Ravindran, A, Ramaswamy, V, on the bottleneck assignment problem, Journal of optimization theory and application 21, 451-458, 1988.

[11] Bhatia, H.L, Time minimization assignment problem, Systems and cybernetics in management 6, 75-83, 1977.

[12] Shalini Arora, Puri, M.C. A lexi search algorithm for a time minimization assignment problem 35(3), 193-213.,1997.

[13] Balakrishna, U, Truncated time - dependent TSP M. Phil, Dissertation , S.V University, Tirupati, India.