Group Assignment Job Constrained Three Dimensional Model

Vijayalakshmi R[#1], Revathi P[#2], Srinivasulu Y[#3] Sundara Murthy M[#4]

[#1]Research Scholar, Dept., of Mathematics, Sri Venkateswara Universty, Tirupati, India
[#]2Assistant professor, Dept., of Mathematics, Brahmmaiah Engineering College, Kovur, Nellore Dist. India
[#3]Research Scholar, Dept., of Mathematics, Sri Venkateswara Universty, Tirupati, India
[#4]Professor Dept., of Mathematics, Sri Venkateswara University, Tirupati, India

***Abstract* - In a classical assignment problem the goal is to find an optimal assignment for agents with tasks without assigning an agent more than once and ensuring that all tasks should complete with minimum cost.**

In this paper we study the problem called "Group Assignment Job Constrained Three Dimensional Model". Let us consider a set of workers W={1,2,……,w}, a set of jobs J={1,2,……,n} and K={1,2,…..,k} represents facilities which influence the cost as a third dimension. The set of W workers due to their identical skills are subdivided into 'p' different groups, i^th^ group is having $w_i = \left|W_i\right|$ workers as a result total workers are w, the set of 'J' jobs are subdivided into 'q' different groups such that in each group have $n_j = \left|J_i\right|$ jobs with total 'n' jobs. Let there are 's' products which require the jobs as components for its finishing. Let the products be called frames and they are F_1, F_2,……,F_s and their frequencies are l_1, l_2,……,l_s . Let 'f_{ij}' be the number of jobs required from j^th^ group for the i^th^ frame then $\sum_{i=1}^{s} f_{ij} \,.\, l_i = n_j^1 \le n_j$ is the number of jobs required for the 's' frames from the j^th^ group of jobs and $n_j^1 \le n_j$. The number of all jobs from the 'q' groups required are $\sum_{j=1}^{q} n_j^{\,1} = n_0 < n$.

There is a restriction that the jobs from same group should use same facility. Where C(i,j,k) is the assignment cost of doing a job in j^th^ group by a worker in i^th^ group using k^th^ facility . The problem is to assign the total number of jobs 'n_0' which are required for the 's' frames subjected to the condition such that the total assignment cost is minimum. A Lexi – Search algorithm is developed using " Pattern ecognition Technique" to get an optimal assignment.

***Key words*- Lexi – Search algorithm, Pattern Recognition Technique, Alphabet table , Word, Search table.**

I. INTRODUCTION

Assignment problem is among the first linear programming problems to be studied extensively. It is a particular case of a transportation problem where the sources are assignees and the destinations are tasks. In a classical assignment problem the goal is to find an optimal assignment of agents to tasks without assigning an agent more than once and ensuring that all tasks are completed.

In this paper we study the problem called "Group Assignment Job Constrained Three Dimensional Model". Let us consider a set of workers W, another set of jobs J. The third dimension which is an independent factor which influences cost is considered as facility which is denoted by K. Again workers W are considered as 'p' groups, i^th^ group is having 'w_i' workers as a result total workers are w, jobs J are considered as 'q' groups such that in each group have n_j jobs with total 'n' jobs and facility K, C(i,j,k), is the cost of doing a job in j^th^ group when assigned to a worker in i^th^ group using facility 'k'.

Let the jobs J={1,2,3, ……. ,n} be grouped as J_1, J_2, ……. , J_q such that J= J_1 U J_2 U…….. U J_q with lJ_jl = n_j , lJl = n, i.e., $n_1 + n_2 +$ ………. $+n_q = n$. Similarly W = {1, 2, …….. , m} be the set of 'm' workers grouped as W_1, W_2, ……… , W_p such that W = W_1 U W_2 U ……… U W_p with lW_il = w_i and lWl = w, that is $w_1 + w_2 +$ …….. $+ w_p = w$.

Let there be S frames that is F_i =(i=1,2,…,s) these frames require various jobs as components. Let f_{ij} be the number of jobs required as components from the jth group of job for the ith frame. Let l_i be the number of frames of the ith frame that is l_i is the frequency of the frame F_i. The total number of jobs required from jth group as components for the S frames along with their frequencies is given by

$$\sum_{i=1}^{s} f_{ij} \cdot l_i = n_j^1 \le n_j$$

That is n^1_j are the total number of jobs from jth group is required for all the frames along with their frequencies. The total number of jobs required for all the frames is given by $\sum_{j=1}^{q} n_j^1 = n_0 < n$ and when jobs are assigned to workers from same group they have to use same facility.

When the cost C(i,j,k) is consider in the process of assignment if $\bar{w}_i$ and $\bar{n}_j$ are respectively the workers in the ith group and jobs in the jth group then the minimum of $\min(\bar{w}_i, \bar{n}_j)$ = α_{ij} will be integer if it is non-zero and in this case the α_{ij} is the number of workers will be assigned to α_{ij} number of jobs with the cost C(i,j,k)* α_{ij} where the kth facility is used and a fixed number of jobs(n_j^1) has to perform from each job group (n_j)

Now our problem is to assign 'n_0' jobs for the workers subject to the conditions such that total assignment cost is minimum. We develop a Lexi-Search algorithm using pattern recognition technique for getting an optimal assignment with total least cost.

II. MATHEMATICAL FORMULATION

Minimize Z (X) = $\sum_{i \in W} \sum_{j \in J} \sum_{k \in K} c(i,j,k).\alpha_{ij}.x(i,j,k)$

……………(1)

Where α_{ij}= Minimum $(\bar{w}_i, \bar{n}_j)$ (i.e., i=1,2,3,….,p & j=1,2,3….,q {Where $(\bar{w}_i, \bar{n}_j)$ are the workers in ith group and Jobs in the jth group unassigned in the process and **α ij** is an integer}

Subject to the constraints

$$\sum_{i}\sum_{j}\sum_{k} \alpha_{ij}\ x(i,j,k) = n_0 < n \quad for\, i = 1,2,...,p;\ j = 1,2,....q,\ k \in K$$

………….. (2)

$$\sum_{s}\sum_{j}\sum_{k} \alpha_{sj}\ x(s,j,k) \le W_s\ ,\ j = (1,2,....,q),\ \ k \in K \qquad ……(3)$$

$$\sum_{i}\sum_{s}\sum_{k} \alpha_{is}\ x(i,s,k) \le n_s\ ,\ i = (1,2,.....,p),\ \ k \in K \qquad ……..(4)$$

$x(i_1, j_1, k_1)$ = $x(i_2, j_2, k_2)$ = 1, where i_1 , i_2 ϵ (1,2,…,p) $j_1=j_2$ & $k_1=k_2$ …………...(5)

$x(i, j, k)$ =0 or 1, i=1,2,….p ; j=1,2,….q & kϵK .... (6)

The constraint (1) is the objective function of the problem i.e., total minimum cost for assigned **n_0** jobs under the given constraints.

The constraint (2) describes the restriction that the total number of assigned jobs n_0, less than n.

The constraint (3) states the number of assigned workers in a group is less than or equal to its capacity.

The constraint (4) shows the number of jobs in a group is less than or equal to its capacity.

The constraint (5) illustrates that same job group when assigned to workers group should use the same facility.

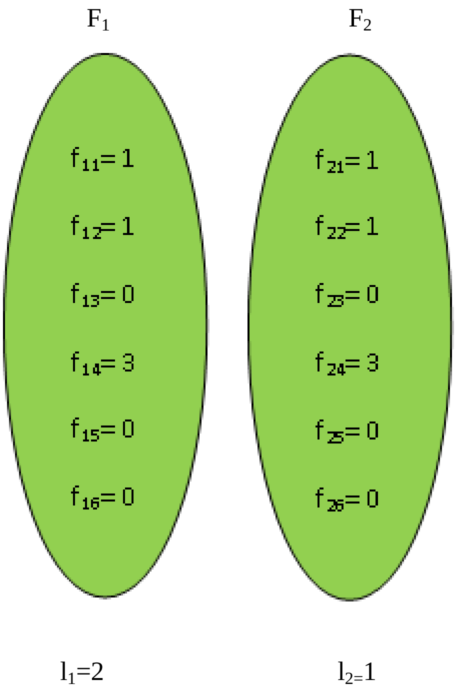
The constraint (6) describes if the ith worker is assigned to jth group of job with kth facility then X(i, j, k) = 1 otherwise 0

III NUMERICAL ILLUSTRATION

The concepts developed will be illustrated by a numerical example for which the total number of workers w = 15 , the total number of jobs n=20 and the facilities k = 2. Again these 20 workers made as 4 groups,15 jobs made as 6 groups i.e.,p=4 and q=6 groups respectively. Let the number of workers in each group be w_1=3, w_2=4, w_3=5, w_4= 3 and jobs as J_1=2, J_2=4, J_3=3, J_4=4, J_5=4, J_6=3. To make l_i frequency of s frames of F a fixed number of jobs has to be assigned from each group that is n_1^1=1, n_2^1 =3, n_3^1=2,n_4^1=3, n_5^1=4, n_6^1=2 and if a job in one set used one of facilities then the remaining jobs in that set should use the same facility. For this problem we took four groups of workers, six groups of jobs with two facilities and the number of jobs to be assigned * are n_1^1+ n_2^1 + n_3^1+n_4^1+ n_5^1+ n_6^1=1+3+2+3+4+2= 15 (i.e.,n_0=15)

Let there be two frames F_1, F_2 and the frequencies of frames are l_1=2 , l_2=1. For these two frames job components from different groups are taken as product component f_{ij}. The following figure-1 represents frames, job components and their frequencies.

FIGURE-1



In the above figure F_1 , F_2 represents frames, l_1=2, l_2=1 represents frequencies of frames. Frame F_1 involves 6 job components and they are represented as n_1^1=0, n_2^1=1, n_3^1=1 n_4^1=0, n_5^1=2, n_6^1=1. f_2 frame involves 6 job components and they are represented as n_1^1=1, n_2^1=1, n_3^1=0, n_4^1=3, n_5^1=0, n_6^1=0. As a result total number of job components in two frames is 15 i.e.,1+3+2+3+4+2=15 also sum of n_i^1 in two frames are satisfying the restriction that they should be equal to the fixed number of jobs assigned from each group. For this problem we took four groups of workers, six groups of jobs using two facilities and the number of jobs to be assigned is fifteen (i.e., n^1=15)

In the following numerical example, C(i, j, k)'s are taken as positive integers but it can be easily seen that this is not a necessary condition. C(i, j, k) means the cost of assigning of that i^{th} worker on j^{th} job with facility k. The following table represents the requirement of the cost to do the job with respect to corresponding worker. Then the cost array C(i, j, k) is given **table-1**.

Table-1

$$C(i, j ,1) = \begin{bmatrix} 01 & 09 & 13 & 21 & -- & 16 \\ -- & 12 & 15 & 18 & 01 & 06 \\ 17 & 10 & -- & 03 & 19 & 11 \\ 05 & 14 & 20 & 05 & -- & 22 \end{bmatrix}$$

Table-2

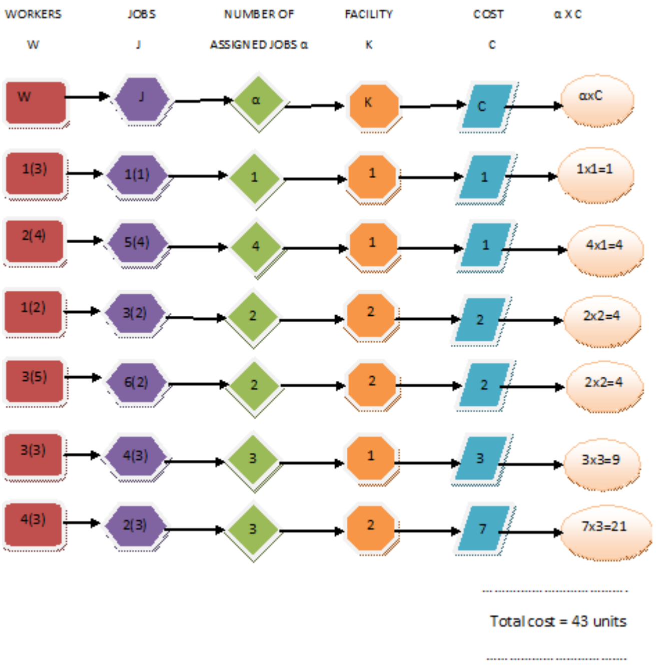
$$C(i, j , 2) = \begin{bmatrix} 18 & 11 & 02 & 15 & 06 & 12 \\ -- & 19 & 13 & 10 & 14 & 09 \\ 13 & 04 & 17 & 20 & 08 & 02 \\ -- & 07 & 12 & -- & 22 & 16 \end{bmatrix}$$

In table-1, C(3, 2, 1) = 10 means that the cost of assigning a job in 2^{nd} group by any individual worker on 3^{rd} group using 1^{st} facility is 10 units.

IV FEASIBLE SOLUTION

Consider an ordered triple set {(1, 1 ,1), (2, 5, 1), (1, 3, 2), (3, 6, ,2), (3, 4, 1), (4, 2,2)} represents a feasible solution mentioned below

FIGURE-2 [FEASIBLE SOLUTION]



The above figure-2, represents a feasible solution. The rectangle shape represents worker group, hexagon shape represents job group, diamond shape represent number of jobs assigned, octagon shape represents facility, parallelogram shape represents the corresponding C(i, j, k) cost and oval shape represents multiple of number of jobs assigned and cost. The values in rectangle indicate group of the worker and in brackets represents number of unassigned workers in that group, values in hexagon indicates group of jobs and in brackets represents the number of unassigned jobs in that group, values in diamond indicates number of jobs assigned , values in octagon indicates facility, value in parallelogram indicates cost and value in oval indicates (number of jobs assigned x cost0f each job).

Here { $w_1(3)$, $j_1(1)$, α(1), k(1), C(1), αxC(1) } represents that $w_1(3)$ means in 1st group number of unassigned workers is 3, $j_1(1)$ means in 1st group number of unassigned jobs is 1 , α(1) means the number of jobs of j_1 assigned to w_1 workers is '1' (which is required) , k(1) represents the facility for performing particular job is '1' ,C(1) represents the cost of that particular job performed by the worker using facility is '1' and αxC(1) represents the product of number of assigned jobs and cost.

SOLUTION PROCEDURE:

In the above **figure-2,** for the feasible solution we observed that there are **6** ordered triples (1, 1 ,1), (2, 5, 1), (1, 3, 2), (3, 6, 2), (3, 4, 1), (4, 2, 2) taken along with the value from the cost matrices in the numerical example in **table-1.** The **6** ordered triples are selected such that they represents a feasible solution according to constraints of mathematical formulation and is represented in **figure-2.** So the problem is that we have to select **6** ordered triples from the cost matrices along with values such that the total cost is minimum and represents a feasible solution. For this selection of 6 ordered triples from cost matrices we arranged 41 ordered triples with the increasing order of their values and call this formation as alphabet table and we will develop an algorithm for the selection of **six ordered triples** along with the checking for the feasibility.

V THE CONCEPTS AND DEFINITIONS

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 feasible solution. The pattern represented in the table-2, is a feasible pattern.

Now V(X) the value of the pattern X is defined as

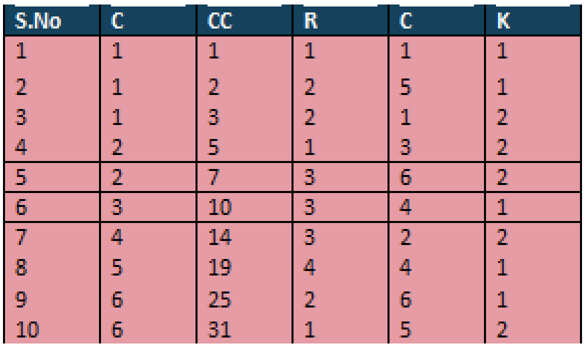
$$V(X) = \sum_{i \in p} \sum_{j \in q} \alpha_{ij} C(i,j,k).X(i,j,k)$$

The value V(X) gives the total cost of the feasible solution represented by X. Thus the value of the feasible pattern gives the total cost 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 the understanding that the other X (i, j, k)'s are zeros.

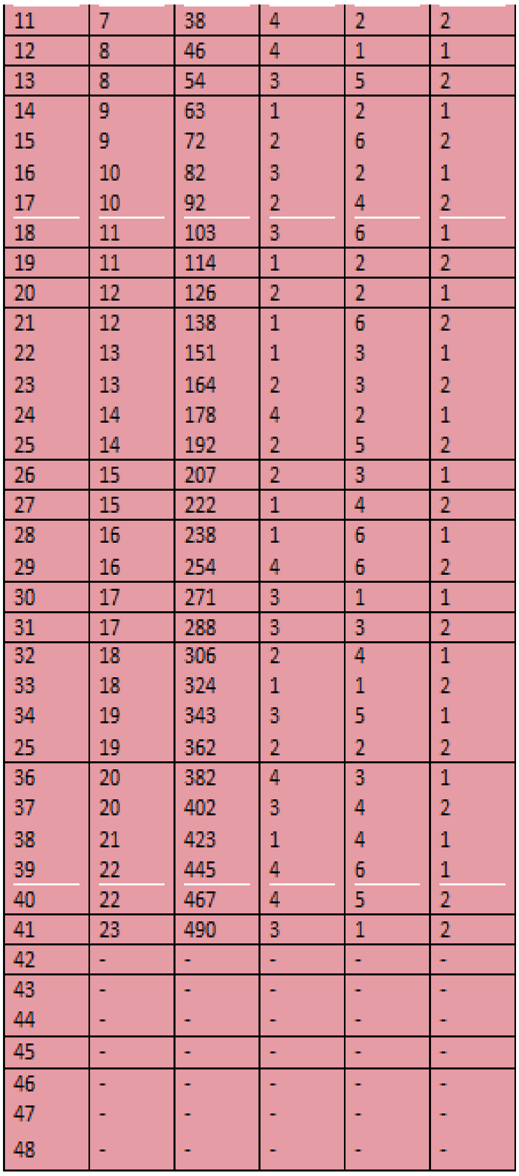
VI ALPHABET TABLE AND A WORD

There are p×q×k ordered triples in the three-dimensional array C. For convenience these are arranged in ascending order of their corresponding costs and are indexed from 1 to pxqxk (Sundara Murthy-1979). Let SN= 1, 2, 3…be the set 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 R, C, K be the array of indices of the ordered triples represented by SN, J and K. CC is the array of cumulative sum of the elements of C. The arrays SN, C, CC, R, C, and K for the numerical example are given in the table-3.
If p∈ SN then (R(p),C(p),K(p)) is the ordered triple and C(a)=C(W(a),J(a),K(a)) is the value of the ordered triple.

Table-3.



S.No	C	CC	R	C	K
1	1	1	1	1	1
2	1	2	2	5	1
3	1	3	2	1	2
4	2	5	1	3	2
5	2	7	3	6	2
6	3	10	3	4	1
7	4	14	3	2	2
8	5	19	4	4	1
9	6	25	2	6	1
10	6	31	1	5	2



11	7	38	4	2	2
12	8	46	4	1	1
13	8	54	3	5	2
14	9	63	1	2	1
15	9	72	2	6	2
16	10	82	3	2	1
17	10	92	2	4	2
18	11	103	3	6	1
19	11	114	1	2	2
20	12	126	2	2	1
21	12	138	1	6	2
22	13	151	1	3	1
23	13	164	2	3	2
24	14	178	4	2	1
25	14	192	2	5	2
26	15	207	2	3	1
27	15	222	1	4	2
28	16	238	1	6	1
29	16	254	4	6	2
30	17	271	3	1	1
31	17	288	3	3	2
32	18	306	2	4	1
33	18	324	1	1	2
34	19	343	3	5	1
25	19	362	2	2	2
36	20	382	4	3	1
37	20	402	3	4	2
38	21	423	1	4	1
39	22	445	4	6	1
40	22	467	4	5	2
41	23	490	3	1	2
42	-	-	-	-	-
43	-	-	-	-	-
44	-	-	-	-	-
45	-	-	-	-	-
46	-	-	-	-	-
47	-	-	-	-	-
48	-	-	-	-	-

Let us consider 7 є SN it represents the ordered triple R(7), C(7), K(7) = (3, 2, 2) then C(7)=4 , CC(7) = 14.

6.1 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}) + D (a_k) x ($\alpha_{i,j,k}$) with V (L_o) = 0 where D (a_k) is the cost array arranged such that D (a_k) < D (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 by L_k, (Sundara Murthy – 1979).

Consider the partial word L_3 = (1, 2, 3)

Then V(L_3) =V(L_2)+ (D (a_k)) (α_{ijk}),

$$\left\{ \begin{array}{l} \text{where}\alpha_{ijk}=\alpha_{251}= \text{minimum } (w_i^1, j_i^1, k) \\ \text{minimum}(w_2^1, j_5^1, 1) = 4 \text{ jobs, } D(a_3) = 1 \end{array} \right\}$$

For example the partial word L_3 = (1, 2, 3) then value of L_3 is V (L_3) = 1+1+1 = 3 and V(x) = 5 and L_3 = V(L_3) = V(L_2)+C(a_k)($\alpha_{i,j,k}$) = 5 + 1 x 0 = 5

6.2 Lower Bound of A partial word LB (L_k):

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

$$LB(L_k) = V(L_K) + D(\ a_{k+1}\ )\ \text{x}\ (n_0^1\ )\ ,$$

{ where n_0^1 represents the number of jobs remaining to be assigned}

Consider the partial word L_3 = (1, 2, 3) and V (L_3) =5

Then LB (L_k) = $V(L_K) + C$ (a_{k+1}) (n_0^1)

LB (L_3) = V(L_3) + (C (a_4) × (6))

= 5 + 2 × 10 = 25

LB (L_k) = V (L_k) + C(a_{k+j})

= V (L_k) + CC (a_k + n - k) - CC (a_k)

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.

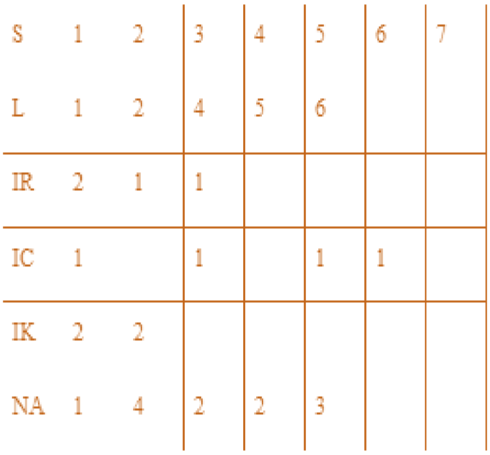
6.3 Feasibility Criterion of a partial word:

An algorithm is developed, in order to check the feasibility of a partial word L_{k+1} = {a_1, a_2...........a_k, a_{k+1}} given that L_k is a feasible word. We will introduce some more notations which will be useful in the sequel.

- **IR** be an array where IR(i) = 1, indicates that a worker of ith group is doing a job , otherwise IR(i)=0.
- **IC** be an array where IC (j) = 1, indicates that in jth group a job is performed by a worker, otherwise IC(j)=0.
- **IK** be an array where IK(i)=1, indicates that facility used by jth group of job
- **L** be an array where L[i] =a_i is the letter in the ith position of a word.Then the values of the arrays IR, IC and L are as follows
- IR (RA) = 1, i = 1, 2, - - - - - , k and IR (j) = 0 for other elements of j. (where RA = R(a_i),CA = (a_i))
- IC (CA) = 1, i = 1, 2, - - - - - , k and IC (j) = 0 for other elements of j
- L (i) = a_i, i = 1, 2, - - - - -, k, and L(j) = 0, for other elements of j
- NA (a_k) = α_{ij} = [w^1(RA),n^1 (CA)] the number of jobs assigned to workers at ith position.

For example consider a sensible partial word L_5 = (1, 2, 4, 5, 6) which is feasible. The array IR, IC, IK and L takes the values represented in table – 4 given below.

TABLE-4

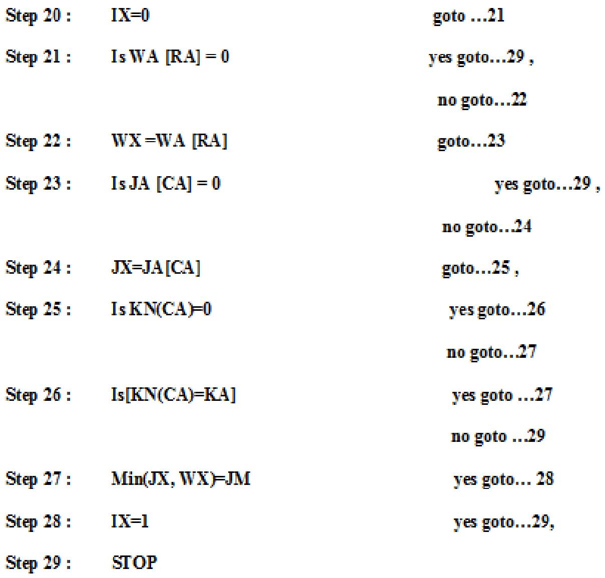


S	1	2	3	4	5	6	7
L	1	2	4	5	6		
IR	2	1	1				
IC	1		1		1	1	
IK	2	2					
NA	1	4	2	2	3		

The recursive algorithm for checking the feasibility of a partial word L_p 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 (a_i), CA=C (a_i).

6.4. ALGORITHMS

ALGORITHM 1: (Algorithm for feasible checking)



```
Step 20 :   IX=0                    goto …21
Step 21 :   Is WA [RA] = 0          yes goto…29 ,
                                    no goto…22
Step 22 :   WX =WA [RA]             goto…23
Step 23 :   Is JA [CA] = 0                 yes goto…29 ,
                                    no goto…24
Step 24 :   JX=JA [CA]              goto…25 ,
Step 25 :   Is KN(CA)=0             yes goto…26
                                    no goto…27
Step 26 :   Is[KN(CA)=KA]           yes goto …27
                                    no goto …29
Step 27 :   Min(JX, WX)=JM          yes goto… 28
Step 28 :   IX=1                    yes goto…29,
Step 29 :   STOP
```

This recursive algorithm will be used as a subroutine in the lexi-search algorithm. We start the algorithm with a very large value, say, 9999 as a trial value of 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 with the partial word L_1= (a_1) = (1). A partial word L_p=L_{p-1} ∗ (a_p) where ∗ indicates chain form or concatenation. We will calculate the values of V (L_p) and LB (L_p) simultaneously. Then two cases arises (one for branching and other for continuing the search).

1. ***LB (L_p) < VT.*** Then we check whether L_p is feasible or not. If it is feasible we proceed to consider a partial word of order (p+1), which represents a sub block of the block of words represented by L_p. If Lp is not feasible then consider the next partial word of order p by taking another letter which succeeds a_p in

the p^{th} position. If all the words Sof order p are exhausted then we consider the next partial word of order (p-1).

2. ***LB (L_P) ≥ VT***. In this case we reject the partial word meaning that the block of words with L_p as leader is rejected for not having an optimal word and we also reject all partial words of order p that succeeds L_p.

Now we are in a position to develop lexi search algorithm to find an optimal feasible word.

6.4. ALGORITHM2: (LEXI-SEARCH ALGORITHM)

The following algorithm gives an optimal feasible word.

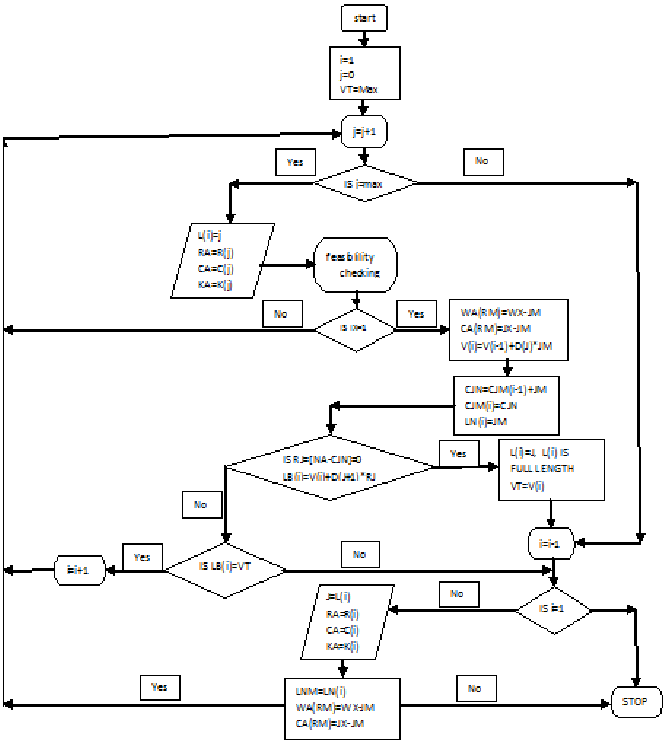
STEP-1: (initialization):

The arrays SN, D, DC, R, C, K, NA(= n_0), WA, JA, KA, max, and VT are made available. RA, CA, KA, L, V, LB, WX, JX, JM, W, N, KN, LN,CJM, LNM and CJN are initialized to zero. The values i=1, j=0.

```
Step1:     j=j+1
           Is (j= max)                              yes goto…2 ,
                                                    no goto…9
Step2 :    L[i]=j
           RA=R(j)
           CA=C(j)
           KA=K(j)                                  yes goto…3
Step 3 :   (CHECK FEASIBILITY OF USING ALGORITHM-1)
           IX=0                                     yes goto…1
                                                    no goto…4
Step 4 :   V(i)=V(i-1)+D(j)*JM                      goto…4a
Step 4a :  WA(RA)=WA(RA)-JM;
           JA(CA)=JA(CA)-JM                         goto…5
Step 5 :   CJN=CJM(i-1)+JM                          goto…5a
Step 5a :  CJM(i)=CJN                               goto…5b
Step 5b :  LN(i)=JM                                 goto …6
Step 6 :   ISRJ=[NA-CJN]=0                          yes goto…8
                                                    no goto …6a
Step 6a :  LB(i)=V(i)+D(j+1)*RJ                     goto…6b,
Step 6b :  LB(i) = VT                               yes goto…7
                                                    no goto … 9
Step 7 :   i=i+1                                    goto …1
Step 8 :   L(i)=j
           L(i) is full length word and is feasible
           VT=V(i), Record VT, L(i)                 goto …9
Step 11:   j=L(i)
           RA=R(j)
           CA=C(j)
           KA=K(j)                                  goto …12
Step 12 :  LNM=LN(i)
           WA(RA) =WA(RA)-LNM;
           JA(CA)=JA(CA)-LNM                        yes goto …1
                                                    no goto … 13
Step 13 :  STOP
```

6.5. *FLOW CHART:*

The flow chart for this algorithm is as follows



Vll *SEARCH TABLE:*

The working details of getting an optimal word, using the above algorithm for the illustrative numerical example are given in the Table-5. The columns (1), (2), (3), (4),……. gives the letters in the first, second, third , fourth so on respectively. The corresponding NA, V and LB are indicated in the next three columns. The rows R, C and K gives the row, column and facility 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-5

S.No	1	2	3	4	5	6	NA	V	LB	R	C	K	REM
1	1						1	1	15	1	1	1	A
2		2					4	5	15	2	5	1	A
3			3							2	1	2	R
4			4				2	9	25	1	3	2	A
5				5			2	13	31	3	6	2	A
6					6		3	22	34	3	4	1	A
7						7				3	2	2	R
8						8				4	4	1	R
9						9				2	6	1	R
10						10				1	5	2	R
11						11	3	43	43	4	2	2	A,VT=43
12					7		3	25	40	3	2	2	A
13						8	3	40	40	4	4	1	A,VT=40
14					8		3	28	46	4	4	1	R,>VT
15				6			3	18	38	3	4	1	A
16					7		2	26	41	3	2	2	R,>VT
17				7			3	21	46	3	2	2	R,>VT
18			5				2	9	33	3	6	2	A
19				6			3	18	38	3	4	1	A
20					7					3	2	2	R
21					8					4	4	1	R
22					9					2	6	1	R
23					10					1	5	2	R
24					11		3	39	55	4	2	2	R,>VT
25				7			3	21	46	3	2	2	R,>VT
26			6				3	14	46	3	4	1	R,>VT
27		3								2	1	2	R
28		4					2	5	27	1	3	2	A
29			5				2	9	39	3	6	2	A
30				6			3	18	46	3	4	1	R,>VT
31			6				3	14	40	3	4	1	R,=VT
32		5					2	5	41	3	6	2	R,>VT
33	2						4	4	15	2	5	1	A
34		3								2	1	1	R
35		4					2	8	26	1	3	2	A
36			5				2	12	33	3	6	2	A
37				6			3	21	37	3	4	1	A
38					7					3	2	2	R
39					8		3	36	42	4	4	1	R,>VT
40				7			3	24	44	3	2	2	R,>VT
41			6				3	17	41	3	4	1	R,>VT
42		5					2	8	35	3	6	2	A
43			6				3	17	41	3	4	1	R,>VT
44		6					3	13	45	3	4	1	R,>VT
45	3						1	1	29	2	1	2	A
46		4					2	5	29	1	3	2	A
47			5				2	9	39	3	6	2	A
48				6			3	18	46	3	4	1	R,>VT
49			6				3	13	45	3	4	1	R,>VT
50		5					2	5	41	3	6	2	R,>VT
51	4						2	4	28	1	3	2	A
52		5					2	8	41	3	6	2	R,>VT
53	5						2	4	43	3	6	2	R,>VT

At the end of the search the current value of **VT = 40** and it the value of the feasible word **L_6=(1,2,4,5,7,8)** it is given in 13th row of the search table – 4 and the corresponding order triples are **(1, 1, 1), (2, 5, 1), (1, 3,2), (3, 6, 2), (3, 2, 2), (4, 4, 1).** For this optimal feasible word the arrays IR, IC, IK, L and NA are given in the following **Table- 6.**

S	1	2	3	4	5	6
L	1	2	4	5	7	8
IR	2	1	2	1		
IC	1	1	1	1	1	1
IK	3	3				
NA	1	4	2	2	3	3

At the end of the search table the optimum solution value of VT is **40** and is the value of optimal feasible word L = (1,2,4,5,7,8). Then the following **figure – 3** represents the optimal solution to the assignment

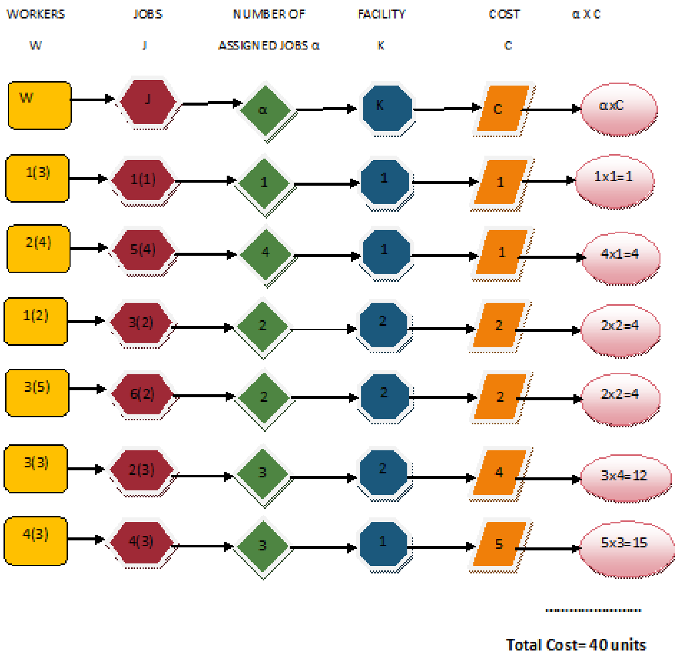


Fig-3 (OPTIMAL SOLUTION)

In figure-4 { $w_1(3)$, $j_1(3)$, α(3), k(1), C(5), αxC(5X3) } represents that $w_1(3)$ means in 1st group of workers number of unassigned workers is 3, $j_1(3)$ means in 1st group of jobs number unassigned jobs are 3 , α(3) means the number of assigned job (which is required) , k(1) is the facility used by that job group ,C(5) is the cost of that job performed by the worker using facility and αxC(5X3=15) is the number of assigned jobs x cost.

According to the pattern represented in **figure-3** is satisfies all the constraints the section 3. The ordered tripled set represents the cost of total number of assigned jobs. The total cost = 1+4+4+4+12+15=40.

VIII *CONCLUSION*

In this chapter, we studied a model of Assignment problem namely Group Asssignment Job Constrained Three Dimensional Model. We have developed a Lexi-Search Algorithm using Pattern Recognition Technique for getting an optimal solution.

REFERENCES:

1. AGARWAL, V. The assignment problem under categorized jobs, *European Journal of Operational Research* 14 (1983) 193-195.
2. AGARWAL, V., TIKEKAR, V.G., HSU, L.F. BOTTLENECK assignment problem under categorization, *Computers and Operations Research* 13(1) (1986) 11-26.
3. BURKARD, R. E. RUDOLF.R, AND WOEGINGER, G.J., Three-dimensional axial assignment problems with decomposable cost coefficients. Technical Report 238, Graz, 1996.
4. CARLOS A., OLIVEIRA, S. AND PANOS M. PARDALOS., Randomized parallel algorithms for the multidimensional assignment problem. *Appl. Numer. Math.*, 49(2004)117–133, 2004.
5. CRAMA, Y. AND F. C. R. SPIEKSMA. Approximation algorithms for three-dimensional assignment problems with triangle inequalities. *European Journal of Operational Research*, 60 (3) (1992) 273–279.
6. FRIEZE, A.M. and J. YADEGAR. An Algorithm for solving 3-Dimensional Assignment Problems with Application to Scheduling a Teaching Practice. J. Ope. Res., 32(1981) (989-995).
7. PANDIT, S.N.N and SUNDARA MURTHY, M. Enumeration of all optimal Job sequence - *Opsearch*, 12, (1975) 35-39.
8. PANNERSELVAM, R. Operations Research second edition, Prentice Hall of India private limited, New Delhi, 2006.

9. PURUSOTHAM, S, SURESH BABU, C. And SUNDARA MURTHY, M. Pattern Recognition based Lexi-Search Approach to the Variant Multi-Dimensional Assignment Problem, *International Journal of Engineering Science and Technology (IJEST).* 3(8) (2011) 6350-6363.

10. SOBHAN BABU.K, CHANDRA KALA.K, PURUSHOTTAM.S, SUNDARA MURTHY.M, A New Approach for Variant Multi Assignment Problem. *International Journal on Computer Science and Engineering*, 02,(05) (2010) 1633-1640.

11. SOMASEKHAR SRINIVAS.V.K **P-AGENTS SEASONAL JOB COMPLETION MODEL**" in the **"International Journal of Scientific Research and Engineering"** for publication. Vol 2 No (11), November,2013.