INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

CONSTRAINED SEASONAL GROUP ASSIGNMENT MODEL

[1]P. Revathi , [2]P.Madhu Mohan Reddy, [3]V.K. Somasekhar Srinivas, [4]E. Sudhakara

[1,3,4]Department of Mathematics, Sri Venkateswara University, Tirupati -517 502, A.P., India
[2]Assistant Professor, Department of Mathematics, Siddharth Institute of Engineering and Technology, Puttur, A.P., India

ABSTRACT- This paper describes a Constrained Seasonal Group Assignment Model, which consists of m machines, n jobs and r seasons. Again, m machines are grouped into p sets of machines, n jobs are grouped into q sets of jobs i.e the number of sets in machines is p and the number of sets in jobs is q. Let the number of elements in each set of machines $M_1, M_2, M_3, \ldots M_p$ be $m_1, m_2, m_3 \ldots m_p$ i.e $|M_i| = m_i$ and the number of elements in each set of jobs $J_1, J_2, J_3, \ldots J_q$ be $n_1, n_2, n_3, \ldots n_q$ i.e $|J_i| = n_i$. Therefore the total number of machines in all sets is taken as m, the total number of jobs is taken as n and the total number of seasons is taken as r. 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. $D(i, j, k)$ The cost/time taken by a machine in M_i for doing job in J_j in the r^{th} season is given. Here third dimension is season. The objective is, to given p sets of machines i.e. M, q sets of jobs i.e. J and r seasons and the three dimensional cost/ time array 'D' , n_0 (<n) jobs are to be assigned with least cost/time subjected to the constraints.

Keywords: Group Assignment, Truncation, Pattern, Lexi Search.

1. INTRODUCTION

Robert and Garfinkel [1] investigated an improved algorithm for bottleneck assignment problem. Barr et al. [2] studied the alternative basis algorithm for Assignment problem. Bhatia [3] investigated time minimizing assignment problem. Shalini and Puri [4] investigated A variant of time minimizing assignment Problem. Shalini and Puri [5] obtained A Lexi Search Algorithm for a Time minimizing assignment problem. Hung and Ram [6] studied the solving the assignment problem by Relaxation. Bertsekas [7] obtained A New Algorithm for the Assignment Problem.

A more general version of cost minimization Assignment Problem is considered by Geetha and Nair [8] obtained a variation of the Assignment problem. Tapadar and Sahu [9] Solving the Assignment Problem using genetic algorithm and simulated anneling. Purusotham [10] studied the problem, namely Pattern Recognition based Lexi-Search Approach to the Variant Multi-Dimensional Assignment Problem. Sobhan Babu et al. [11] studied A

new Approach for Variant Multi Assignment Problem. Madhu Mohan Reddy et al. [12] studied the Constrained Three Dimensional Job Assignment Model.

2. PROBLEM DESCRIPTION

In this paper, we study the problem called "Constrained Seasonal Group Assignment Model". For this we took m machines, n jobs and r seasons. Again, m machines are grouped into p sets of machines, n jobs are grouped into q sets of jobs i.e the number of sets in machines is p and the number of sets in jobs is q. Let the total number of elements in all the sets of machines is m, the number of elements in all the sets of jobs n and the total number of seasons is taken as r. Further we can write as the following:

Symbolically, in this model we consider p sets of machines and q sets of jobs. The sets of machines M_1, $M_2, M_3, \ldots M_p$ are considered such that $M = M_1 \cup M_2 U M_3 \cup \ldots \cup M_p$ and $|M_i| = m_i$. Here m_{1+} m_{2+} $m_{3+\ldots+}m_p = m$. Similarly q set of jobs $J_1, J_2, J_3, \ldots J_q$ are considered such that $J = J_1 \cup J_2 \cup J_3 \cup \ldots J_q$ and $|J_i| = n_i$, Here $n_1 + n_2 + n_3 + \ldots + n_q = n$ and the season set S={1,2,…r}.

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. $D(i, j, k)$ The cost/time taken by a machine in M_i for doing job in J_j in the r^{th} season is given. Here third dimension is season. In a feasible assignment the total assigned number of machines in a particular case of machine set (M_s) is less than or equal to that number m_s and the total assigned number of jobs in a particular case of job set (J_s) is less than or equal to that number n_s, which is necessary constraint to the feasibility. If a job is done on a machine in i^{th} season, the same machine is available on j^{th} season where $i \neq j$ i.e. different jobs can be done on the same machine in different seasons.

In the feasible assignment schedule all the subset of machines should be represented. i.e. in the feasible solution there is at least one machine from each of the p sets of machines and all the subset of jobs should be represented. i.e. in the feasible solution there is at least one job from each of the q sets of jobs.

The objective of the problem is, given p sets of machines i.e. M, q sets of jobs i.e. J and r seasons and the three dimensional cost/ time array 'D' , n_0 (<n) jobs are to be assigned with least cost/time subjected to the constraints.

3. MATHEMATICAL FORMULATION

$D(i, j, k)$ Means that cost of a machine in M_i is used by a job in Jj for doing in the k^{th} season. From n jobs we want to assign n_0 jobs. Here n_0 less than n.

$$Minimize Z(X) = \sum_i \sum_j \sum_k D(i, j, k).x(i, j, k) \quad \ldots \quad (1)$$

here i∈ (1,2,3…p), j∈ (1,2,…q), k∈ S= {1, 2, 3,…r}

Subjected to constraints

$$\sum_i \sum_j \sum_k x(i, j, k) = n_0 \quad \text{i∈ (1,2,3…p), j ∈ (1,2,…q), k∈ S={1,2,…r}} \quad \ldots (2)$$

Here $n_0 < n$

$$\sum_i \sum_j \sum_k x(i,j,k) \le m_s, i \in M, j \in J, k \in S \quad \ldots (3)$$

$$\sum_i \sum_j \sum_k x(i,j,k) \le n_s, j \in J_s \quad \text{here} \quad i \in M, k \in S \quad \ldots (4)$$

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

$$\sum_{i=1}^{p} MI(i) = p \quad \ldots (5)$$

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

$$\sum_{i=1}^{q} JI(i) = q \quad \ldots (6)$$

x(i,j,k)=0 or 1, here $i \in (1,2,3,\ldots p)$, $j \in (1,2,3,\ldots q)$, $k \in S$ ... (7)

The constraint **(1)** is the objective function which measures the minimum cost 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 n.

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

The constraint **(4)** describes in the job assignment schedule, the number of jobs assigned from the set J_s should be less than or equal to its number n_s.

The constraint **(5)** describes, in the assignment schedule, all the subset of machines are involved. i.e. in the feasible solution there is at least one machine from each of the p sets of machines.

The constraint **(6)** describes in the assignment schedule, all the subset of jobs are involved. i.e. in the feasible solution there is at least one worker from each of the r sets of jobs.

The constraint **(7)** describes that the i^{th} set in M is used by j^{th} set in J for doing the k^{th} season in S then x (i, j, k) =1 otherwise 0.

We presented the Pattern Recognition Technique based Lexi Search Algorithm (LSA) for this model. We tested the proposed algorithm by different set of problems.

4. Numerical Illustration

The concept and algorithm developed by a numerical example for which the number of machine sets are p=4 and in each set the machines are m_1 =2, m_2 =1 m_3=3,m_4 =4. Therefore the total number of machines are $m_1+m_2+m_3+m_4$=2+1+3+4=10=m. i.e. the total number of machines=10. Similarly the number of jobs sets q=5 and in each set the jobs are J_1 =3, J_2 =1, J_3=2, j_4 =3, J_5=2 . Therefore the total number of jobs are $J_1+J_2+J_3+J_4+J_5$=3+1+2+3+2=11=q. i.e. the total number of jobs=11 and the number of seasons r=2

D (i, j, k) means the cost/time that a machine in i^{th} set working on a machine j^{th} set in k^{th} season. The table-1 represents the requirement of the cost for doing the job with respect to corresponding machine on the particular season. Then the cost array D (i, j, k) is given below.

TABLE-1

D(i,j,1)=

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

D(i,j,2)=

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

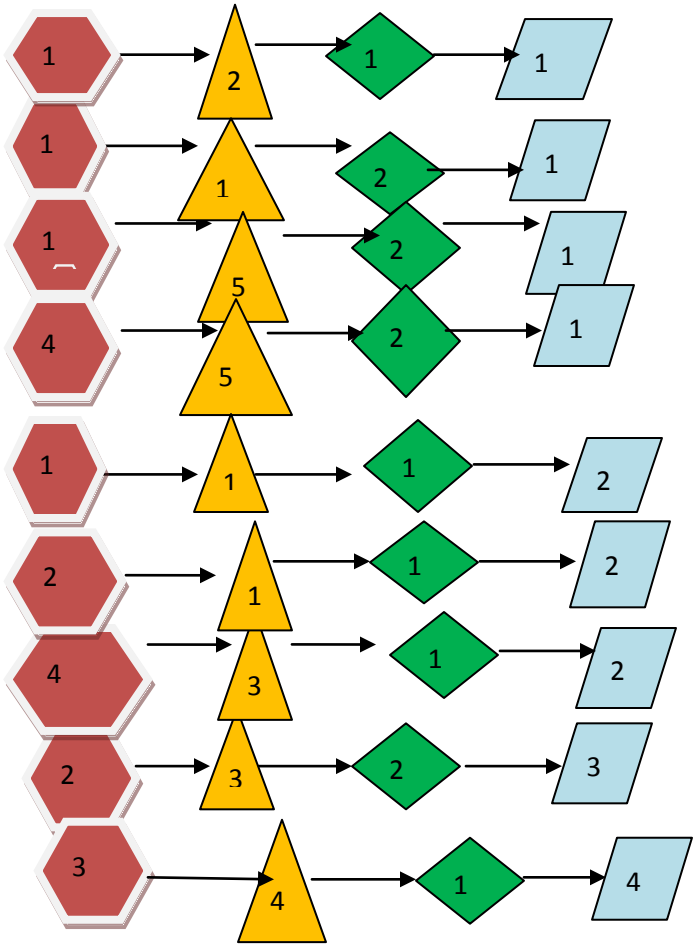
In table-1, D (2, 4, 2) = 9 means that the cost of a machine in 2nd set for assigned a job in 4th set in kth season is 9. For the table-1 we developed alphabet table and search table. The total number of jobs should be assigned $n_0 < n$. The objective of the problem is total assigned cost/distance should be minimum.

5. CONCEPTS AND DEFINITION

5.1. Feasible solution

The figure-**1** represents a feasible solution. The hexagon shapes represent machines, triangle shapes represent jobs, diamond shapes represent seasons and parallelogram shapes represent the corresponding cost of machine, job and season. The values in hexagon indicates name of the machine, values in triangle indicate name of the job and values in diamond shapes indicate name of the season.

Figure-1



From the figure-1, first group machine is assigned to second group job in first season with one unit cost, first group machine is assigned to first group job in second season with one unit cost and so on.

According to the pattern represented in **figure-1** is satisfied all the constraints in the section 3. It is a feasible solution. The corresponding ordered tripled set represents the cost

D(1,2,1)=1, D(1,1,2)=1,D(1,5,2)=1, D(4,5,2)=1,D(1,1,1)=2, D(2,1,1)=2, D(4,3,1)=2, D(2,3,2)=3, D(3,4,1)=4.

The total cost=1+1+1+1+2+2+2+3+4=17.

Solution Procedure: In the above figure-1 for the feasible solution we observe that there are 9 ordered triples taken along with the value from the cost matrices for the numerical example in table-1. The 9 ordered triples are selected such that they represent a feasible solution in fig-1. So the problem is that we have to select 9 ordered triples from the cost matrices (4×5×2) along with values such that the total value is minimum and represents a feasible solution. For this selection of 9 ordered triples we arrange the 4×5×2=40(maximum) ordered triples in the increasing order of the costs and call this formation as alphabet table and we will develop an algorithm for the selection of ordered triples along with the checking for the feasibility.

5.2 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. Now V(X) the value of the pattern X is defined as V (x) = $\sum\sum\sum$ D (i, j, k) X (i, j, k)

The value V(X) gives the total cost/ time of the assignment for the 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.

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

Table-2

$$X(i,j,1)=\begin{pmatrix}1&1&0&0&0\\0&0&0&0&0\\0&0&0&1&0\\0&0&1&0&0\end{pmatrix} \qquad X(i,j,2)=\begin{pmatrix}1&0&0&0&1\\0&0&1&0&0\\0&1&0&0&0\\0&0&0&0&1\end{pmatrix}$$

5.3 Alphabet Table

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

Alphabet Table (Table-3)

SN	D	CD	M	J	T
1	1	1	`1	2	1
2	1	2	1	1	2
3	1	3	1	5	2
4	1	4	3	2	2
5	1	5	4	5	2
6	2	7	1	1	1
7	2	9	2	1	1
8	2	11	4	3	1
9	2	13	1	4	2
10	**2**	**15**	**2**	**5**	**2**
11	3	18	1	3	1
12	3	21	2	5	1
13	3	24	1	3	2
14	3	27	3	3	2
15	3	30	4	3	2
16	4	34	2	3	1
17	4	38	3	4	1
18	4	42	3	1	2
19	4	46	3	5	2
20	4	50	4	4	2
21	5	55	4	1	1
22	5	61	4	5	1
23	5	66	1	2	2
24	5	71	2	1	2
25	5	76	3	4	2
26	6	82	1	5	1
27	6	88	3	2	1
28	6	94	4	4	1
29	6	100	2	2	2
30	7	107	2	2	1
31	7	114	3	3	1
32	7	121	4	2	2
33	8	129	4	2	1
34	9	138	1	4	1
35	9	147	3	5	1
36	9	156	2	4	2

Let us consider $10 \in SN$. It represents the ordered triple (M(10),J(10),T(10))=(2,5,2).Then D(10)= 2 and CD(10)=15.

5.4. 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 $L_\alpha = (a_1, a_2, - - - - , a_\alpha)$ can be defined as follows.

$LB(L_\alpha) = V(L_\alpha) + C(a_{\alpha+j}) = V(L_\alpha) + CD(a_\alpha + n_0 - k) - CD(a_\alpha)$

Consider the partial word

$$L_4 = (4, 7, 10, 18) = V(L_4) = 1+2+2+4 = 09$$

$$LB(L_4) = V(L_4) + CD(a_4 + n_0 - \alpha) - CD(a_4)$$

$$= 09 + CD(18 + 9 - 4) - CD(18)$$

$$= 09 + CD(23) - CD(18)$$

$$= 09 + 47 - 34 = 24$$

5.5. ALGORITHMS

Algorithm 1: (Checking for the Feasibility)

Step 0: IX=0 GOTO 2

Step 2: IS (IM (RA)) +1 ≤ m_{RA} IF YES GOTO 4

IF NO GOTO 20

Step 4: IS (IJ (CA)) +1 ≤ n_{CA} IF YES GOTO 6

IF NO GOTO 20

Step 6: IS (MX (RA) = 0 IF YES, MNA=MN (I-1) +1

GOTO 8

IF NO GOTO 8

Step 8: IS (JX (CA) = 0 IF YES, JNB=JN (I-1) +1
GOTO 10

IF NO GOTO 10

Step 10: Is N_0 -I ≥ max (p - MNA, q - JNB)

IF YES GOTO 12

IF NO GOTO 20

Step 12: 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 big value say '∞' as a testing 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 solution. 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)

The arrays SN, D, CD, M, J, T, n, n_0, p, q and r are made available. IM, IJ, IT, MX, JX, TX, L, V 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) =D (J)

MN (I) =1

JN (I) =1,

TN (I) =1 GOTO 3B

IF NO GOTO 3A

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

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

Step 4 : IS (LB ≥VT) IF YES GOTO 11

IF NO GOTO 5

Step 5: RA=R (J)

CA=C (J)

KA=T (J) GOTO 5A

Step5A: IS I=1 IF NO GOTO6

IF YES GO TO 5B

Step5B: IM (RA) =1

IJ (CA) =1

IK (TA) =1 GOTO 10A

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

L (I) is full length word and is feasible

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

Write [R(L(J)), C(L(J)),T(L(J))] ,Here $J=1,2,\ldots N_0$ GOTO 13

Step 10: IM (RA)=IM(RA)+1

IJ (CA)=IJ(CA)+1

IK (TA)=IK(TA)+1

MN(I)= MNA

JN(I)=JNB

TN(I)=TNC GOTO 10A

Step10A: 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), KA=T(J)

IM (RA)=IM(RA)-1

IJ (CA)=IJ(CA)-1

IK (TA)=IK(TA)-1 GOTO 2

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.

5.6. Search Table

The working details of getting an optimal word, by using the above algorithm for the illustrative numerical example are given in the table-4. The column R, C, T and remarks are acceptability of the partial words. In the following table A indicates ACCEPT and R indicates REJECT.

Table-4

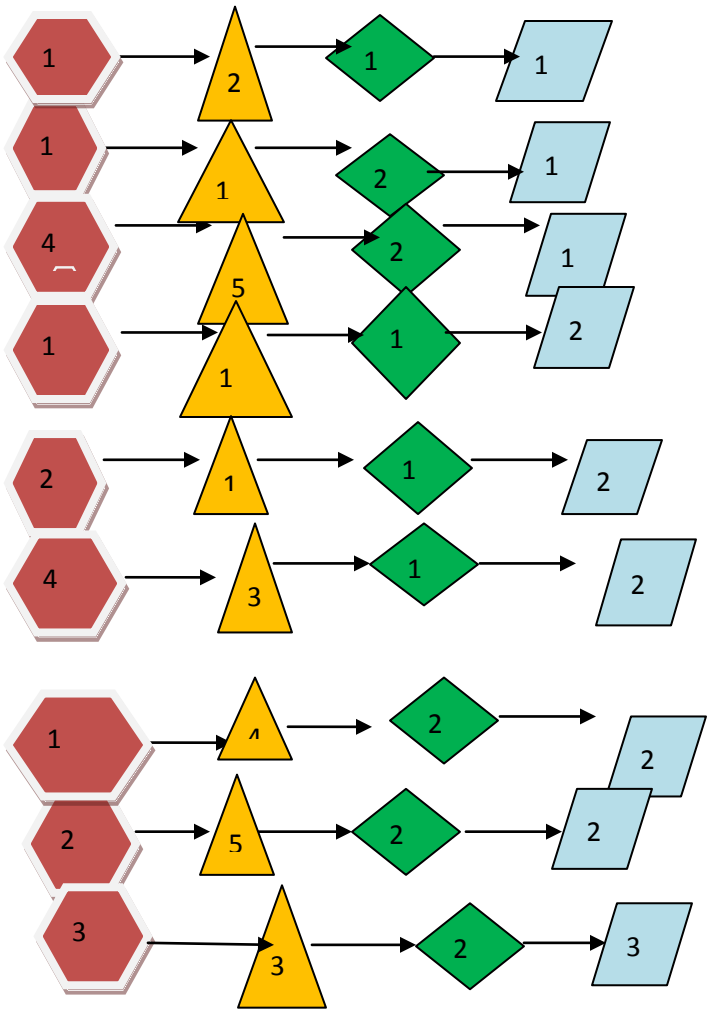
S.N.	1	2	3	4	5	6	7	8	9	V	LB	R	C	T	RE
1	1									1	13	1	2	1	A
2		2								2	13	1	1	2	A
3			3							3	13	1	5	2	A
4				4						4	13	3	2*	2	R
5				5						4	14	4	5	2	A
6					6					6	14	1	1	1	A
7						7				8	14	2	1	1	A
8							8			10	14	4	3	1	A
9								9		12	14	1*	4	2	R
10								10		12	15	2	5*	2	R
11								11		13	16	1*	3	1	R
12								12		13	16	1*	5	1	R
13								13		13	16	1*	3	2	R
14								14		13	16	2	3	2	A
15									15	16	16	3	3	2*	R
16									16	17	17	2*	3	1	R
17									**17**	**17**	**17**	**3**	**4**	**1**	**A=VT**
18								15		13	17	4	3	2	R=VT
19							9			10	15	1*	4	2	R
20							10			10	16	2*	5	2	R
21							11			11	17	1	3	1	R=VT
22						8				8	15	4	3	1	A
23							9			10	15	1*	4	2	R
24							10			10	16	2*	5	2	R
25							11			11	17	1	3	1	R=VT
26						9				8	16	1*	4	2	R
27						10				8	17	2	5	2	R=VT
28					7					6	15	2	1	1	A
29						8				8	15	4	3	1	A
30							9			10	15	1*	4	2	R
31							10			10	16	2	5*	2	R
32							11			11	17	1	3	1	R=VT
33						9				8	16	1*	4	2	R
34						10				8	17	2	5	2	R*
35					8					6	16	4	3	1	A
36						9				8	16	1*	4	2	R
37						10				8	17	2	5	2	R=VT

38					9					6	17	1	4	2	R=VT
39				6						5	16	1	1	1	A
40					7					7	16	2	1	1	A
41						8				9	16	4	3	1	A
42							9			11	16	1*	4	2	R
43							10			11	17	2	5	2	R=VT
44						9				9	17	1	4	2	R=VT
45					8					7	17	4	3	1	R=VT
46				7						5	17	2	1	1	R=VT
47			4							3	14	3	2*	2	R
48			5							3	16	4	5	2	A
49				6						5	16	1	1	1	A
50					7					7	16	2	1	1	A
51						8				9	16	4	3	1	A
52							9			11	16	1	4	2	A
53								10		13	16	2	5	2	A
54									11	16	16	1*	3	1	R
55									12	16	16	1*	5	1	R
56									13	16	16	1*	3	2	R
57									**14**	**16**	**16**	**3**	**3**	**2**	**A,VT=16**
58								11		14	16	1	3	1	R=VT
59							10			11	17	2	5	2	R>VT
60						9				9	17	2	4	2	R>VT
61					8					7	17	4	3	1	R>VT
62				7						5	17	2	1	1	R>VT
63			6							4	18	1	1	1	R>VT
64		3								2	14	1	5	2	A
65			4							3	14	3	2*	2	R
66			5							4	17	4	5	2	R(>VT)
67		4								2	16	3	2	2	R(>VT)
68	2									1	14	1	1	2	A
69		3								2	14	1	5	2	R
70			4							4	15	3	2	2	A
71				5						5	15	4	5	2	R
72				6						7	18	1	1	1	R
73			5							4	17	4	5	2	R
74		4								3	17	3	2	2	R
75	3									1	16	1	5	2	R

At the end of the search the current value of VT is (01+01+01+02+02+02+02+02+03) = 16 and it is the value of the feasible word L_9 = (1, 2, 5, 6, 7, 8, 9, 10, 14) it is given in 57th row of the search table – 4 and the corresponding order triples are (*1, 2, 1), (1, 1, 2), (4, 5, 2) (1, 1, 1) , (2, 1, 1),*(4,3,1),(1,4,2),(2,5,2)(3,3,2)**.**

The following **figure-2** represents an optimal solution.

Figure-2



From the figure-2, The hexagon shapes represent machines, triangle shapes represent jobs, diamond shapes represent seasons and parallelogram shapes represent the corresponding cost of machine, job and season. The value in hexagon indicates name of the machine, value in triangle indicates name of the job and value in diamond shapes indicates name of the season and first group machine is assigned to second group job in first season with one unit cost, first group machine is assigned to first group job in second season with one unit cost and so on.

According to the pattern represented in **figure-2** is satisfied all the constraints of the section 3. The ordered tripled set represents the cost D(1,2,1)=1,D(1,1,2)=1,D(4,5,2)=1,D(1,1,1)=2,D(2,1,1)=2,D(4,3,1)=2,D(1,4,2)=2,D(2,5,2)=2,(3,3,2)=3. The total cost=1+1+1+2+2+2+2+2+3=16.

6. CONCLUSIONS

In this paper, we have studied a model namely **Constrained Seasonal Group Assignment Model**. We have developed a new algorithm which is efficient, accurate and easy to understand. In this assignment problem it deals with groups of machines, groups of jobs and seasons. So it is a three dimensional group assignment problem. Though it is a group assignment problem, we still treat it as a zero one programming problem. i.e. One job in a sub group assigned to one machine in a sub group and season. Identical machines/jobs are in one group. In this

grouping it reduces the size of the problem, i.e., for example if it is a 50 machines, 40 jobs and 3 seasons problem, it can be conveniently grouped as 5×4×3 instead of 50×40×3,when identical machines and jobs are involved. Interestingly still we treat it as zero one programming problem. i.e., one job is assigned to the one machine only. The algorithm is discussed in detail with the help of a numerical illustration. We tested the proposed algorithm by different set of problems.

Acknowledgements

The authors would gratefully acknowledge the valuable advice from Prof. M. Sundara Murthy, Sri Venkateswara University, Tirupati.

REFERENCES

[1] Robert, S. and Garfinkel. Technical Note—An Improved Algorithm for the Bottleneck Assignment Problem. *Operations Research* 19(7), 1971, 1747-1751.

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

[3] Bhatia. H.L. Time minimizing assignment problem. *Systems and Cybernatics inmanagement* 6, 1977, 75-83.

[4] Shalini, Arora and Puri, M.C. A variant of time minimizing assignment Problem, *European Journal of Operational Research*, 1997, 314-325.

[5] Shalini Arora, Puri, M.C. A Lexi Search Algorithm for a Time minimizing assignment proble. *European Journal of Operational Research* 35(3), 1998, 193-213.

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

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

[8] Geeta, S. and Nair, K.P. A variation of the Assignment problem, *Euro.J. Of O*r,68, 1993, 422-426.

[9] Tapadar Rudrajit and Sahu Anshuman. Solving the Assignment Problem using genetic algorithm and simulated anneling, *IJAM*, 36(1), 2007

[10] Purusotham, S. Suresh Babu, C., Sundara Murthy, M. Pattern Recognition based Lexi-Search Approach to the Variant Multi-Dimensional Assignment Problem,*International Journal of Engineering Science and Technology*, Vol. 3 (8), 2011, 6350-6363.

[11] Sobhan Babu. K Sobhan babu.K.,chandra kala.K., Purushottam.S. and Sundara Murthy, M. A new Approach for Variant Multi Assignment Problem, *International Journal on Computer Science and Engineering*,Vol. 02, No. 05, 2010, 1633-1640

[12] Madhu Mohan Reddy, P., Suresh Babu, C., Somasekhar Srinivas, V.K. and Sundara Murthy, M. Constrained three dimensional job assignment maodel, *InternationalJournal of Engineering Sciences and Engineering Sciences and research technology*, Vol.-2(9), 2013, 2361-2387.