Three Dimensional P- Trucks Route Minimum Cost Supply to The Head Quarter From Cities

Revathi P[1], Suresh Babu C[2], Purusotham S[3], Sundara Murthy M[4]
Research Scholar, Dept. of Mathematics, S. V. University, Tirupati, Andhra Pradesh, India.
Academic Consultant, Dept. of Mathematics, S. V. University, Tirupati, Andhra Pradesh, India.
Asst. Professor, Statistics and OR Division,VIT University, Vellore, Tamil Nadu, India
4.Professor, Dept. of Mathematics, S. V. University, Tirupati, Andhra Pradesh, India.

ABSTRACT:
Many Combinatorial programming problems are NP-hard (Non Linear Polynomial), and we consider one of them called P-path minimum cost connectivity to head quarter{1}from the cities. Let there be n cities and the cost matrix D (i, j, k) is given from i^{th} city to j^{th} city using k^{th} facility. There can be an individual factor which influences the distances/cost and that factor is represented as a facility k. We consider m<n cities are in cluster and to connect all the cities in subgroup (cluster) from others by using same facility k. From different cities to Head Quarter city in P-path the supply of load will be according to the requirement. The problem is to find minimum cost to connect all the cities to head quarter (say 1) through p-paths with the requirements of the load and subject to the above considerations. For this problem we developed a Pattern Recognition Technique based Lexi Search Algorithm.

KEYWORDS: *Lexi Search Algorithm, Pattern Recognition Technique, Partial word, Pattern, Mathematical formation, load*

I. INTRODUCTION:

In recent years the development of networks in the area of telecommunication and computer has gained much importance. One of the main goals in the design process is to reach total connectivity at minimum cost/distance. The total connections are in some paths (say P). Similar problems arise in the planning of road maps, integrated circuits. The technical restriction that the number of connections at a node is bounded is modelled by introducing constraints that bound the node degrees. Garey at al [2] proved that the resulting degree-constrained minimum. In this paper we studied a variation of Minimum spanning models. For this we developed a Lexi- algorithm based "Pattern Recognition Technique" to solve this problem.

Some of the researchers studied variations in the Minimum Spanning Tree (MST) problems. They are Pop, P.C [6], Karger [3] found a linear time randomized algorithm based on a combination of Boruvka's algorithm and the reverse-delete algorithm. The problem can be solved deterministically in linear by Chazelle [1] .Its running time is o (m, α(m,n)) where function α grows extremely slowly. Thus Chazelle's algorithm takes very close to linear time. Seth Pette [4, 5] have found a probably optimal deterministic comparison-based minimum spanning tree algorithm. The general network design, given a directed graph G(N,A) where N={1,2,.....n}denotes the set of nodes and the set of arcs, each arc (i,j)€ A , various algorithms are available in the literature to find the Shortest Path between one or any pair of nodes,. For example Dijkstra's algorithm solves the single-pair, single-source, problem if edge weights may be negative. Floyd-Warshall algorithm solves all pair's shortest paths. Perturbation theory finds (at worst the locally shortest path).The path finding is applicable to many kinds of networks, such as roads, utilities, water, electricity telecommunications and computer networks alike, the total number of algorithms that have been developed over the years is immense. A selected cross-section of approaches towards path finding and the related fields of research, such as transportation. GIS, network analysis, operation research, graph theory, artificial intelligence and robotics, to mention just a few examples where path finding theories all are employed. Even though the different research literature tends to group the types of shortest paths problems slightly different, one can discern, in general, between paths that are calculated as one-to-one, one-to-some, one-to-all, all-to-one and all-to-all shortest paths.

Sobhan Babu [8] studied a variation of spanning models using pattern recognition technique [9]. Let there be N cities to be connected to the headquarter city {1}. There is an individual factor which influences the distances/cost and that factor is represented as a facility K. If the city j_1 and j_2 different cities are connected from city i_1 then k_1 and k_2 should be same. Suresh Babu [10] studied another variation of spanning models, which is reverse case of [8]. The problem is to find optimal solution for all the cities connected to head quarter {1} with minimum cost by using k facility.

II. LEXICOGRAPHIC SEARCH USING PATTERN RECOGNITION TECHNIQUE:

Lexicographic Search Approach is a systematized Branch and Bound approach, developed by Pandit in the context of solving of loading problem in 1962. In principle, it is essentially similar to the Branch and Bound method as adopted by Little et.al-1963. This approach has been found to be productive in many of the Combinatorial Programming Problems. It is significance mentioning that. Branch and Bound can be viewed as a particular case of Lexicographic Search approach {Pandit-1965]. The name Lexicographic Search itself suggests that, the search for an optimal solution is done in a systematic manner, just as one searches for the meaning of a word in a dictionary and it is derived from Lexicography the science of effective storage and retrieval of information. This approach is based on the following grounds [Pandit -1963].

(i) It is possible to list all the solutions or related configurations in a structural hierarchy which also reflects a hierarchical ordering of the corresponding values of these configurations.

(ii) Effective bounds can be set to the values of the objective function, when structural combinatorial restraints are placed on the allowable configurations.

The basic principle is described as follows [Rajbhongshi-1982].Consider set of symbols. A= (1,2,3,....., n) and the different possible sequences of length k of these symbols. Thus (α_1, α_2, ...α_k) is a k-word, formed from the alphabetic order on the elements of A, We will be able to define a unique ordered list of words of length not exceeding m, where m is finite. Words of length $k \leq m$ are called incomplete words standing for the set or block of the (m- k).Words of length k Searching for an optimum word is la problem of finding the word of minimum value (in the case of a minimizing problem). In the Lexi search defined by the solution of the problem. The search efficiency of a Lexi Search algorithm is based in this approach depends on the choice of an appropriate Alphabet-Table, where two conflicting characteristics of the search list have to be taken into account; one is the difficulty in setting bounds to the values of the partial words (that defines partial solutions representing subsets of solutions). The other difficulty is checking the feasibility of a partial word. Thus we get two situations in the choice of the alphabet-table [Sundara Murthy-1979].By this method, in this problem we get Computation of lower bound is easy, while the feasibility checking is difficult. When the process of feasibility checking of a partial word becomes difficult and the lower bound computation is easy, a modified Lexi- Search i.e. Lexi- Search with recognising the Pattern of the Solution known as Pattern Recognition Technique which was the main efficiency of the algorithm, first the bounds are calculated and then the partial word, for which the value is less than the initial trial value are checked for the feasibility. The Pattern-recognition technique can be described as follows.

"A unique pattern is associated with each solution of a problem. Partial pattern defines a partial solution. An alphabet-table is defined with the help of which the words, representing the pattern are listed in a Lexicographic order. During the search for an optimal word, when a partial word is considered, first bounds are calculated and then the partial words for which the value is less than the trail value are checked for the feasibility"

Using Pattern Recognition technique reduces the dimensions requirement of the problem. For this problem find an optimal solution X which is a three dimensional array, the problem can be reduced to a linear form of finding an optimal word of length n. This reduction in the dimension for some problems reduces the computational word in getting an optimal solution [Sundara Murthy – 1979, Vidyulata – 1992, Ramana and Uma shankar -1995]. The present paper uses the Lexicographic Search in general and makes use of the Pattern Recognition present paper uses the Lexicographic Search in general and makes use the Pattern Recognition approach.

III. PROBLEM DESCRIPTION:

In this paper we study the problem called "Three dimensional P- Trucks route minimum cost supply to the Head Quarter {1} from cities". The objective is to find the minimum total cost to supply required capacities through P trucks in P paths from each city to head quarter city {1}. Each path requirement is not greater than 'α' units of capacity. The total capacity is 'Pα' units. Let there be N= {1, 2, 3, 4……n} cities whose costs N *x N x K* are given. In this chapter we consider Pattern Recognition Technique based Lexi-Search Approach for minimum spanning network connectivity problem (msncp). There is a restriction that M is subset of N cities must have same facility. An exact algorithm is proposed for this minimum spanning network connectivity problem (msncp). The algorithm solves the problem by identify the key patterns which optimize the objective of the cost/distance.

Let N be the set of n stations defined as N={1,2,3,4,……..n} and the set of q facilities in K = {1,2, . . ., q}. Let D (i, j, k) be the cost from i^{th} city to j^{th} city using k^{th} facility where i, j Є N and kЄ K. Let there are set of m cities in M= {1, 2, 3... m} such that M be the cluster and M is subset of N. We want to connect all the (n-1) cities to head quarter city by P-paths. Each city connected to head quarter city {1} either directly or indirectly. The P- trucks starts at different cities in P paths and has to reach to head quarter city {1}.The total capacity in P- trucks (Pα) is greater than or equal to the availability of the cities. In each city there is availability of some material (TL(i)). The objective of the problem is to find minimum cost to connect all the n-1 cities having availability capacity of load at each city supply to head quarter city {1}. For this we developed an algorithm called as Lexi-Search algorithm using pattern recognition Technique and it is illustrated with a suitable numerical example for three paths.

IV. MATHEMATICAL FORMULATION:

$$Minimize\ Z(X) = \sum_{I=1}^{n}\sum_{J=1}^{n}\sum_{K=1}^{q} D(i,j,k), X(i,j,k) \dots\dots\dots\dots (1)$$

Subject to constraints.

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{q} X(i,j,k) = n-1 \dots\dots\dots (2)$$

$$\sum_{s=1}^{p} X(1, n_{s1}) = P \text{---------- (3)}$$

$1,\alpha_{11},\alpha_{12}\ldots..\alpha_{1n1}$ be the cities in the first path to the Head Quarter city,

$1,\alpha_{i1},\alpha_{i2}\ldots\ldots\alpha_{ini}$ (i=1,2,…P) be the cities in the i^{th} path with n_{i+1} cities.(cities taken in reverse order)

$$\sum_{s=1}^{ni-1} X(\alpha_{is},\alpha_{is+1}) = n_i\text{-}1 \text{---------- (4)}$$

$$\sum_{s=1}^{ni} TL(\alpha_{is}) \leq \alpha \text{---------- (5)}$$

Let i_1, i2ЄM,

If $X(i_1,j_1,k_1)=X(i_2,j_2,k_2)=1$. Then $k_1 = k_2$ --------- (6)

X (i, j, k) = 0 or 1..... (7)

Equation (1) represents the objective of them problem i.e. to find minimum total distance from the cities to head quarter city. Equation (2) represents total number of connections in the network. The equation (3) represents the total number of paths from cities to head quarters (say p). Equation (4) represents the total number of cities connected in each path in reverse order. Equation (5) represents load/ capacity in each path.Equation (6) represent that the total cities in M using same facility. Equation (7) describes that if a city 'i' is connected to city 'j' using facility 'k' then X(i,j,k) = 1. Otherwise it will be equal to '0'.

V. NUMERICAL FORMULATION:

The concept and algorithm developed will be illustrated by a numerical example for which total number of cities N = {1, 2, 3, 4, 5, 6, 7, 8, 9}.In each path the capacity of load (say α) has to supply from cities to Head Quarter city. The number of paths is 3. Among them the cities 2,4&7 are taken as separate cluster say

M = {2, 4, 7}. All the cities in M have connected to other cities should be used same facility. Then the distance matrices D (i , j, k) are as follows.

TABLE-1

	∞	∞	∞	∞	∞	∞	∞	∞	∞
	01	∞	21	-	25	-	09	28	31
	19	05	∞	11	03	18	27	-	01
	31	28	-	∞	16	-	11	26	32
D(i,j,1)	17	28	15	06	∞	30	14	24	20
	23	13	-	-	22	∞	-	04	33
	29	10	12	20	02	-	∞	32	16
	21	27	25	08	-	23	09	∞	-
	18	-	03	-	26	-	17	3	∞

TABLE-2

	∞	∞	∞	∞	∞	∞	∞	∞	∞
	26	∞	11	-	-	09	29	14	10
	07	15	∞	02	-	-	24	01	-
	25	07	01	∞	-	15	13	22	29
D(i,j,2)	03	22	-	19	∞	-	28	10	19
	17	13	-	02	30	∞	-	32	12
	14	04	-	20	16	-	∞	27	31
	01	-	03	08	33	18	-	∞	23
	24	05	21	-	12	-	-	06	∞

In the above table (1&2) the value distances d(i, i, k), where (i=1,2,...n) as '∞' and d(i,j,k), where (j=1,2,...n) as '-' indicates the disconnectivity between cities. Here the entire D(i,j,k) are taken as positive integers and in this numerical example we are given 9 cities. Suppose D (2, 6, 2) =09 means the distance of the connecting the city 2 to 6 by using facility 2 is 09. .For our convenience the total cities in cluster M identified by the array B as follows in Table-3.

Table-3

cites	1	2	3	4	5	6	7	8	9
Cluster	0	1	0	1	0	0	1	0	0

In the above numerical example given in Table-3, B (i) =1 means that the city i belongs to cluster M, Otherwise B (i) = 0. Suppose B (2) =1, city 2 is cluster M.

Table-4

LOAD

Q	1	2	3	4	5	6	7	8	9
	-	30	30	40	50	70	60	40	40

From the above table-4, Q (i) = β means Load requirement of city j is β. suppose

Q (2) = 30 means that the requirement of load at city 2 is 30. Here Q (1)= '-' means that the city 1 act as head quarter has no load 1.Each path has capacity of load to supply from cities in that path to Head Quarters is **α= LD =150** units . The total number of paths (P) is 3.

VI. CONCEPT AND DEFINITIONS:

6.1 Definition of a pattern: An indicator three-dimensional arrayX which is associated the an assignment is called a "pattern". A pattern is said to be feasible if X is solution.

$$V(X) = \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} D(i,j,k)\ X(i,j,k)$$

The pattern represented is a feasible pattern. The value V(X) 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 X(i, j, k)=1, which understanding that the other X(i, j, k) 's are zeros.

6.2 Feasible Solution: Consider the ordered pairs { (2,1,1), (3,9,2), (4,2,1), (5,1,2), (6,4,2), (7,5,1), (8,1,2), (9,8,2) } represents the pattern given in the tables 5 & 6, which is a feasible solution.

Table-5

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

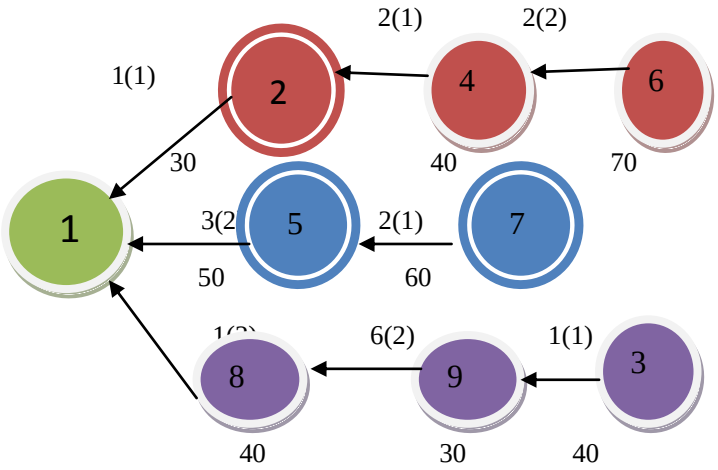
Table-6

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

The above solution X (2, 1, 1) =1, represents that city 2 is connected to city 1 using facility 1. In similar way X(5,1,2) represents city 5 is connected to city 1 by facility 2 and all the cities are connected directly or indirectly to head quarter 1. So the above solution gives a feasible solution shown in the figure-1.

In following figure-1, the values in circles indicates name of the cities. Also values at each arc in parenthesis and before parenthesis represent the facility and distance between the respective two nodes. The value below the circles represents the availability of load (capacity) to that particular city.

Figure-1



The above figure-1 represents a feasible solution. In first path the city 4 is connected by the city 6 using facility 2, city 2 is connected by city 4 using facility 1, city 1 is connected by city 2 using facility 1. In second path, the city 5 is connected by city 7 using facility 1, city 1 is connected by city 5 using facility 2. In third path city 9 is connected by city 3 using facility 1, city 8 is connected by city 9 using facility 2, city1 is connected by city 8 using facility 2. Hence the solution of the above pattern X is as follows

Z= D{(2,1,1),(3,9,1),(4,2,1),(5,1,2),(6,4,2),(7,5,1),(8,1,2),(9,8,2)}

=1+1+2+3+2+2+1+6= 18**.**

6.3 Definition of Pattern:

An indicator three-dimensional array which is associated with connection is called a 'pattern'. A Pattern is said to be feasible if X is a solution.

$$V(X) = \sum_{i\epsilon N}\sum_{j\epsilon N}\sum_{k\epsilon K} D(i,j,k)X(i,j,k)$$

The value V(x) is gives the total cost of the tour for the solution represented by X. The pattern represented in the tables-5&6 is a feasible pattern. The value V(X) gives the total distance of the network for the solution represented by X. Thus X is the feasible pattern gives the total distance 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 triple (i, j, k) for which X (i, j, k)=1, with understanding that the other X(i,j,k) are zeros. The ordered triple set{(2,1,1),(3,9,1),(4,2,1),(5,1,2),(6,4,2),(7,5,1),(8,1,2),(9,8,2)} represents the pattern given in table – 4&5, which is feasible solution.

6.4: ALPHABET TABLE:

There are **n×n×k** ordered triples in the three-dimensional array X. For convenience these are arranged in ascending order of their corresponding cost and are indexed from 1 to M (Sundara murthy-1979). Let SN= {1, 2, 3 ...} be the set of indices. Let D be the corresponding array of cost/Distance. If a,b€SN and a≤b then D(a)≤D(b). Also let the array R, C, K be the array of row, column and facility indices of the ordered triple represented by SN and CD be the array of cumulative sum of the elements of D. For convenience same notation D and K are used for the cost array and facility array in the alphabet table. The array S, D, CD, R, C, and K for the numerical example is given in the table-7. If p€SN then (R(p),C(p),K(p)) is the ordered triple and D(a)= T(R(a), C(a),K(a))is the value of the ordered triple and CD(a) =$\sum_{i=1}^{a} D(i)$

TABLE-7

SN	D	CD	R	C	K
1	01	01	2	1	1
2	01	02	3	9	1
3	01	03	3	8	2
4	01	04	4	3	2
5	01	05	8	1	2
6	02	07	4	2	1
7	02	09	7	5	1
8	02	11	3	4	2
9	02	13	6	4	2
10	03	16	3	5	1
11	03	19	9	3	1
12	03	22	5	1	2
13	03	25	8	3	2
14	04	29	6	8	1
15	04	33	7	2	2
16	05	38	3	2	1
17	05	43	9	2	2
18	06	49	5	4	1
19	06	55	9	8	2
20	07	62	3	1	2
21	07	69	4	2	2
22	08	77	8	4	1
23	08	85	8	4	2
24	09	94	2	7	1
25	09	103	8	7	1
26	09	112	2	6	2
27	10	122	7	2	1
28	*10*	*132*	*2*	*9*	*2*
29	10	142	5	8	2
30	11	153	3	4	1
31	11	164	4	7	1
32	11	175	2	3	2
33	12	187	7	3	1
34	12	199	6	9	2
35	12	211	9	5	2
36	13	224	6	1	2
37	13	237	4	6	2
38	13	250	6	2	2
39	14	264	5	7	1

40	14	278	2	8	2
41	14	292	7	1	2
42	15	307	5	3	1
43	15	322	3	2	2
44	15	337	4	6	2
45	16	353	4	5	1
46	16	369	7	9	1
47	16	385	7	5	2
48	17	402	5	1	1
49	17	419	9	7	1
50	17	436	6	1	2
51	18	454	3	6	1
52	18	472	9	1	1
53	18	490	8	6	2
54	19	509	3	1	1
55	19	528	5	4	2
56	19	547	5	9	2
57	20	567	5	9	1
58	20	587	7	4	1
59	20	607	7	4	2
60	21	628	2	3	1
61	21	649	8	1	1
62	21	670	9	3	2
63	22	692	6	5	1
64	22	714	4	8	2
65	22	736	5	2	2
66	23	759	6	1	1
67	23	782	8	6	1
68	23	805	8	9	2
69	24	829	5	8	1
70	24	853	3	7	2
71	24	877	9	1	2
72	25	902	2	5	1
73	25	927	8	3	1
74	25	952	4	1	2
75	26	978	4	8	1
76	26	1004	9	5	1
77	26	1030	2	1	2
78	27	1057	3	7	1
79	27	1084	8	2	1
80	27	1111	7	8	2
81	28	1139	2	8	1
82	28	1167	5	2	1
83	28	**1195**	5	**7**	**2**
84	29	1224	7	1	1
85	29	1233	2	7	2
86	29	1262	4	9	2
87	30	1292	5	6	1
88	30	1322	9	8	1
89	30	1352	6	5	2
90	31	1383	2	9	1
91	31	1414	4	1	1
92	31	1445	7	9	1
93	32	1477	4	9	1
94	32	1509	7	8	1
95	32	1541	6	8	2
96	33	1607	6	9	1
97	33	1607	8	6	2

Let us consider 92 Є SN .It represents the ordered triple (R(92),C(92),K(92))=(7,9,1).

Then D (92) = 31 and CD(92) = 1445.

6.5 Definition of the Alphabet –Table and the Word: Let SN= (1,2,...) be the set of indices.CD be an array of cumulative sums of elements in D. Let arrays R,C and K be respectively, the row, column and facility indices of the ordered triples. Let $L_k = \{a_1, a_2, ..., a_k\}$, a_i €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 alphabet order as (1,2,....,) and the ordered sequence L_k is defined as a "word "of length k. A word L_k is called a "sensible word". If $a_i \leq a_{i+1}$, for i=1,2,....k-1 and if this condition is not met it is called a "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 letter. The condition for feasibility of a word requires the existence of at least a word and the feasibility of the word defines the position of the Leader.

6.6: Value of the Word: Let the value of the (partial) word defined as L_k .$V(L_k)$ is defined recursively.as$V(L_k)=V(L_{k-1})+V(a_i)$ with $V(L_0)=0$, where $D(a_i)$ is the distance array arranged such that $D(a_i) \leq D(ai_{+1})$. 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_4= (Sundara Murthy-1979).

Consider a partial word L_4= (1, 2, 5, 6)
Then V (L_4) = 1+1+1+2=5

6.7: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 defines as

$$LB(L_k) = V(L_K) + \sum_{j=1}^{n-k} D(a_{k+1}) = V(L_k) + DC(a_k + n - 1 - k) - DC(a_k)$$

Consider the partial word L_4 = (1,2,5,6)

Then $V(L_4)$ =1+1+1+2=5

$LB(L_k)=V(L_k)+DC(a_4+n-1-k)-DC(a_k)$

=5+DC(6+8-4)-DC(6)=5+DC(10)-DC(6)=5+16-7=14.

This is obtained by concatenating the first (n-k) letters of SN to the partial word.

6.8: Feasibility criterion of partial word:

An algorithm was 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, i∈ N indicates that the i^{th} city is connected to some city j. Otherwise IR (i) = 0
- IC be an array where IC (j) = 1, j ∈ N indicates that the i^{th} city is connected from some city i. Otherwise IC (j) = 0
- M be an array, where M (i) = N_i, indicates that the i^{th} city is belongs to N_i cluster, otherwise M (i) = 0.
  SW be an array where SW (i) = j indicates that the i^{th} city is connected to some city j, Otherwise SW (i) = 0
- LW be an array where L[i] = α_i, i∈ N □ is the letter in the i^{th} position of a word.

The values of the arrays IR, IK, SW and LW are as follows

IR (R (a_i)) = 1, i = 1, 2, - - - - - , k and IR (j) = 0 for other elements of j.

IC (C (a_i)) = 1, i = 1, 2, - - - - - , k and IC (j) = 0 for other elements of j.

SW(R (a_i)) = C ((a_i)), i = 1, 2, - - - , k and SW (j) = 0 for other elements of j.

LW (i) = N_i, i = 1, 2, - - - - -, k, and LW (j) = 0 for other elements of j.

The recursive algorithm for checking the feasibility of a partial word L_k is given as follows. For this algorithm we have TR = R (a_{k+1}), TC = C (a_{k+1}) and TK = K (a_{k+1}). We start with the partial word L_1 = (a_1) = (1). A partial word L_k is constructed as $L_k = L_{k-1} * (\alpha_k)$. Where * indicates chain formulation. We will calculate the values of V (L_k) and LB (L_k) simultaneously. Then two situations arises one for branching and 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 under (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 p by taking another letter which succeeds a_k in the position. If all the words of order p are exhausted then we consider the next partial word of order (k-1).

2. LB (L_k) $\geq$ 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.

- For example consider a sensible partial word L_4 = (1, 2, 5, 6) which is feasible. The array IR, IC, L, SW takes the values represented in table - 8 given below.

TABLE-8

	1	2	3	4	5	6	7	8	9
L	1	2	5	6					
IR	1	1	1	1	1			1	1
IC	2	1	1				1	1	
SW		1	9	2				1	1

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 TR=R (a_i), TC=C (a_i).

VII. ALGORITHMS:

ALGORITHM 1: (Algorithm for feasibility checking)

STEP 0:	IX=0	GO TO 1
STEP1:	IS (TC=HC)	IF YES GOTO 2 IF NO GOTO 3
STEP 2:	IS (PA $\leq$ P)	IF YES GOTO 4 IF NO GOTO 16
STEP 3:	IS (IC [TC] =1)	IF YES GOTO 16 IF NO GOTO 4
STEP 4:	IS (IR [TR] =1)	IF YES GO TO 16 IF NO GOTO 5
STEP 5:	Z=P-PA RP=N-1-I IS RP $\geq$ Z	IF YES GO TO 6 IF NO GO TO 16
STEP 6:	W=TR	GOTO 7
STEP 7:	IS (SW [TR] = =0)	IF YES GOTO 15 IF NO GOTO 8
STEP 8:	W=SW [W]	GOTO 9
STEP 9:	IS (W= TC))	IF YES GOTO 16 IF NO GOTO 7

STEP10:	IS (b [TR] = =1)	IF YES GO TO 11 IF NO GOTO 15
STEP11:	IS NM = 0	IF YES GO TO 12 IF NO GOTO 13
STEP 12:	A = b [TR] F [A] = TK	GO TO 14
STEP 13:	IS F (A) = TK	IF YES GOTO 14 IF NO GOTO 16
STEP 14	NM = NM +1	GOTO 15
STEP 15:	IX=1	
STEP 16:	STOP	

$L_k = L_{k-1}$ * Where * indicates chain formulation. We will calculate the values of V (Lp) and LB (Lp) simultaneously. Then two situations 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 processed to consider a partial word of under (P+1). Which represents a sub-block of the block of Words represented by L_p

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

ALGORITHM 2: (Lexi -Search Algorithm)
STEP 1: (Initialization)

The arrays SN, D, CD, R, C and K, the values of n, M, LD,P,MAX are made available, IR,IC,SW, L, V, LB are initialized to zero. The values I=1, J=0, PA =0, ST (TL) = 0, VT= ∞ .

STEP 2**:**	J=J+1 ST (TL) = 0 IS (J>MAX)	IF YES GOTO 15 IF NO GOTO 3
STEP3**:**	L (I) =J TR= R (J) TC= C (J) TK= T (J)	GOTO 4
STEP4:	V (I) =V (I-1) +D (J) LB (I) =V (I) +CD ((J+N-1-I)-CD (J))	GOTO 5
STEP5:	IS (LB (I)>=VT)	IF YES GOTO 11 IF NO GOTO 6
STEP 6:	(CHECK FEASIBILITY BY USING ALGORITHM 1) IS IX =0	IF YES GO TO 2 IF NO GO TO 7
STEP 7:	IS (I= n-1)	IF YES GOTO 8 IF NO GOTO 12
STEP 8:	ST (TL) =ST (TL) + Q (TR)	GO TO 9
STEP 9:	IS ST (TL) < LD	IF YES GOTO 10 IF NO GO TO 2
STEP 10:	W = SW (TR)	GOTO 11
STEP11:	IS (SW [TR] = =0)	IF YES GOTO 14 IF NO GOTO 8
STEP12:	L [I] =J IR [TR] =1 SW [TR] =TC IS (TC= HC)	IF YES {PA=PA+1}, GOTO 13 IF NO {IC [TC] =1}, GOTO 13
STEP 13:	I=I+1	GOTO2
STEP 14:	VT=V [I] L [I] =J	GOTO 15
STEP 15:	I=I-1	GOTO 16

STEP 16: J=L [I]
TR=R [J]
TC=C [J]
IR [TR] =0
SW [TR] =0
L [I+1] =0
IS (TC= HC) IF YES {PA = PA -1, GOTO 17}
IF NO {IC [TC] =0, GOTO 17}

STEP 17: ST (TL) =ST (TL) - Q (TR) GO TO 18

STEP 18: W = SW (TR) GOTO 19

STEP19: IS (SW [TR] = =0) IF YES GOTO 20
IF NO GOTO 17

STEP 20: IS (b [TR] = 1) IF YES GOTO 21
IF NO GOTO 2

STEP 21: NM=NM-1 GOTO 22

STEP 22: IS (I= 1) IF YES GOTO 23
IF NO GOTO 15

STEP 23: STOP

$L_k = L_{k-1}$ * Where * indicates chain formulation. We will calculate the values of V(Lp) and LB (Lp) simultaneously. Then two situations arises one for branching and 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 processed to consider a partial word of under (k+1). Which represents a sub-block of the block of Words represented by LP

2. LB (L_k) ≥VT . In this case we reject the partial word LK. 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 p that succeeds L_k.

VIII. SEARCH TABLE:

The working detail of getting an optimal word using the above algorithm for the illustrative numerical example is given in the following table-9. The columns named (1), (2),(3).....gives the letters in the first, second, third and so on places respectively. The columns R, C,K gives the row, column, facility and load indicates by 'TL'. The last column gives the remarks regarding the acceptability of the partial word. In the following table 'A' indicates ACCEPT and 'R' indicates REJECT.

TABLE- 9

SL.NO	1	2	3	4	5	6	7	8	V	LB	R	C	K	Q	REMARKS
1	1								1	11	2	1	1	30	A
2		2							2	11	3	9	1	70	A
3			3						3	11	3	8	2		R
4			4						3	12	4	3	2		R
5			5						3	14	8	1	2	40	A
6				6					5	14	4	2	1	70	A
7					7				7	14	7	5	1	110	A
8						8			9	14	3	4	2		R
9						9			9	15	6	4	2	110	A
10							10		12	15	3	5	1		R
11							11		12	15	9	3	1		R
12							12		12	15	5	1	2	50	A
13								13	15	15	8	3	2		R
14								14	16	16	6	8	1		R
15								15	16	16	7	2	2		R
16								16	17	17	3	2	1		R
17								17	17	17	9	2	2		R
18								18	18	18	5	4	1		R
19								19	18	18VT	9	8	2		A
20							13		12	16	8	3	2		R
21							14		13	17	6	8	1		R

22							15		13	18	7	2	2		R,=VT
23						10			10	16	9	3	1		R
24						11			10	16	9	3	1		R
25						12			10	17	5	1	2	50	A
26							13		13	17	8	3	2		R
27							14		14	18	6	8	1		R,=VT
28						13			10	18	8	3	2		R,=VT
29					8				07	15	3	4	2		R
30					9				07	16	6	4	2	110	A
31						10			10	16	3	5	1		R
32						11			10	16	9	3	1		R
33						12			10	17	5	1	2		A
34							13		13	17	8	3	2		R
35							14		14	18	6	8	1		R,=VT
36						13			10	18	8	3	2		R,=VT
37					10				08	17	3	5	1		R
38					11				08	18	9	3	1		R,=VT
39				7					05	15	7	5	1	110	A
40					8				07	15	3	4	2		R
41					9				07	16	6	4	2	110	A
42						10			10	16	3	5	1		R
43						11			10	16	9	3	1		R
44						12			10	17	5	1	2	50	A
45							13		13	17	8	3	2		R
46							14		14	18	6	8	1		R,=VT
47						13			10	18	8	3	2		R,=VT
48					10				08	17	3	5	1		R
49					11				08	18	9	3	1		R,=VT
50				08					05	16	3	4	2		R
51				09					05	17	6	4	2	110	A
52					10				08	17	3	5	1		R
53					11				08	18	9	3	1		R,=VT
54				10					06	19	3	5	1		R,>VT
55			6						04	16	4	2	1	70	A
56				7					06	16	7	5	1	110	A
57					8				08	16	3	4	2		R
58					9				08	17	6	4	2	110	A
59						10			11	17	3	5	1		R
60						11			11	17	9	3	1		R
61						12			11	18	5	1	2		R,=VT
62					10				09	18	3	5	1		R,=VT
63				8					06	17	3	4	2		R
64				9					06	18	6	4	2		R,=VT
65			7						04	17	7	5	1	110	A
66				8					06	17	3	4	2		R
67				9					06	18	6	4	2		R,=VT
68			8						04	18	3	4	2		R,=VT
69		3							02	12	3	8	2	70	A
70			4						03	12	4	3	2		R
71			5						03	14	8	1	2	40	A
72				6					05	14	4	2	1	70	A
73					7				07	14	7	5	1	110	A
74						8			09	14	3	4	2		R
75						9			09	15	6	4	2	110	A
76							10		12	15	3	5	1		R
77							11		12	15	9	3	1	70	A
78								12	15	15VT	5	1	2	50	A
79							12		12	15	5	1	2		R,=VT
80						10			10	16	3	5	1		R,>VT
81					8				07	15	3	4	2		R,=VT
82				7					05	15	7	5	1		R,=VT

83			6						04	16	4	2	1		R,>VT
84		4							02	14	4	3	2		R
85		5							02	16	8	1	2		R,>VT
86	2								01	12	3	9	1	70	A
87		3							02	12	3	8	2		R
88		4							02	14	4	3	2		R
89		5							02	16	8	1	2		R,>VT
90	3								01	14	3	8	2	70	A
91		4							02	14	4	3	2		R
92		5							02	16	8	1	2		R,>VT
93	4								01	16	4	3	2		R,>VT

At the end of the search the current value of VT is 15 and it is the value of the optimal feasible word L_8 = (1, 3, 5, 6, 7, 9,11, 12). It is given in the 78th row of the search table. The arrays IR, IC, SW, SWI and LW take the values represented in the table-10 given below. Hence the pattern gives the optimal feasible word.

TABLE-10

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
L	1	3	5	6	7	9	11	12	
IR		1	1	1	1	1	1	1	1
IC	3	1	1	1	1			1	
SW		1	8	2	1	4	5	1	3

From the above arrays optimal feasible word L_8 = (1, 3, 5, 6, 7, 9, 11, 12) and the respective ordered triplets{(2,1,1),(3,8,2),(4,2,1),(5,1,2),(6,4,2),(7,5,1),(8,1,2),(9,3,1)} represents the pattern given in the table-11 &12.

Table-11

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

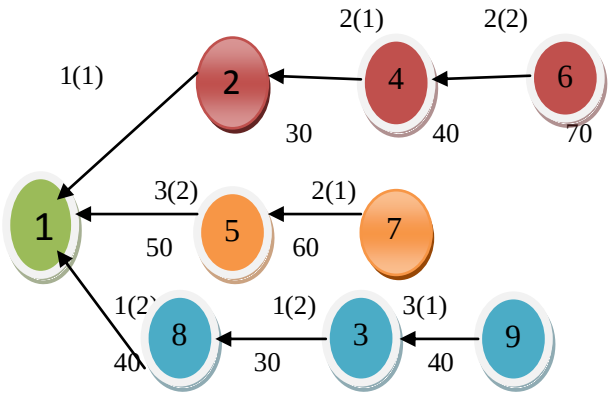
Table-12

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

The paths represented by the above pattern is {(2,1,1),(3,8,2),(4,2,1),(5,1,2),(6,4,2),(7,5,1),(8,1,2),(9,3,1)}.The cities {2,4,7}used the same facility1.The diagrammatic representation of this solution can also see in figure-2.

In following figure-2, the values in circles indicates name of the cities. Also values at each arc in parenthesis and before parenthesis represent the facility and distance between the respective two nodes. The value below the circles represents the availability of load (capacity) to that particular city.

Figure-2



The above figure-2 represents an **optimum feasible** solution. In first path the city 4 is connected by the city 6 using facility 2, city 2 is connected by city 4 using facility 1, city 1 is connected by city 2 using facility1.In second path, the city 5 is connected by city 7using facility 1, city 1 is connected by city 5 using facility 2. In third path city 3 is connected by city 9 using facility 1, city 8 is connected by city 3 using facility 2, city1is connected by city 8 using facility 2. Hence the solution of the above pattern X is as follows

Z= D{(2,1,1),(3,8,2),(4,2,1),(5,1,2),(6,4,2),(7,5,1),(8,1,2),(9,3,1)}

=1+1+2+3+2+2+1+3= 15

IX. CONCLUSION:

In this paper, we developed an **Alphabet Table** and **Lexi- search Algorithm** by using pattern recognition technique for solving this model. The model is formulated as a Zero-One programming problem. By using a suitable numerical example to understand the concepts and steps involved in the algorithm to find an **optimum solution** with the given constraints. Based on this experience we can say that this **algorithm** is applicable for higher dimensions also and more over it is very efficient. For this paper References given below.

REFERENCES:

[1]. A Minimum Spanning Tree Algorithm with Inverse-Ackermann Type Complexity, B. Chazelle, Journal ACM 47 (2000), 1028-1047. Prelim. Version in FOCS 1997.

[2]. Garey, Michael R.; Johnson, David S. (1979), Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, ISBN 0-7167-1045-5.

[3]. Karger, David R.; Klein, Philip N.; Tarjan, Robert E. (1995), "A randomized linear-time algorithm to find minimum spanning trees", *Journal of the Association for Computing Machinery* **42** (2): 321–328, doi:10.1145/201019.201022, MR 1409738

[4]. Pettie, Seth; Ramachandran, Vijaya (2002), "A randomized time-work optimal parallel algorithm for finding a minimum spanning forest", *SIAM Journal on Computing* **31** (6): 1879–1895, doi: 10.1137/S0097539700371065, MR 1954882.

[5]. Pettie, Seth; Ramachandran, Vijaya (2002), "Minimizing randomness in minimum spanning tree, parallel connectivity, and set maxima algorithms", *Proc. 13th ACM-SIAM Symposium on Discrete Algorithms (SODA '02)*, San Francisco, California, pp. 713–722.

[6]. Pop, P.C."New models of the Generalized Minimum Spanning Tree Problem", Journal of Mathematical Modeling and Algorithms, Volume 3, issue 2, 2004, 153-166.

[7]. Reeves, C.R., "Modern Meta heuristics Techniques for Combinatorial Problems", Blackwell, Oxford, 1993.

[8]. Sobhan Babu, K., Chandra Kala, K., Purusotham, S. and Sundara Murthy, M. "A New Approach for Variant Multi Assignment Problem", International Journal on Computer Science and Engineering, Vol.02, No.5, 2010, 1633-1640.

[9]. Sundara Murthy, M. "Combinatorial Programming: A Pattern Recognition Approach," A Ph.D. Thesis, REC, Warangal. 1979.

[10]. Suresh Babu C, Sobhan Babu K and Sundara Murthy M, 2011, published a paper entitled "Variant Minimum Spanning Network Connectivity Problem" by the International Journal of Engineering Science and Technology for publication. Volume 3, No. 1, Jan-Feb 2012