Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 29s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Higher Dimensional P - Path Route Minimum Distance Supply To The Destinations From The Central Station

Madhu Mohan Reddy P[1], Jayalakshmi P[2] , Rajani Kanth V[3]

[1]Associate Professor,Department of Mathematics, Sreenivasa Institute of Technology and Management Studies,India-517127

Email ID: mmrphdsv@gmail.com

[2] Associate Professor,Department of Mathematics, Siddharth Institute of Engineering &Technology,India517583

Email ID: jayachetu85@gmail.com

[3]Professor, Department of Computer Science and Engineering, Sreenivasa Institute of Technology and Management Studies,India517127

Email ID: drrkinskit@outlook.com

Cite this paper as: Madhu Mohan Reddy P, Jayalakshmi P,Rajani Kanth V (2025) Higher Dimensional P - Path Route Minimum Distance Supply To The Destinations From The Central Station *Journal of Neonatal Surgery*, 14 (29s), 879-891

ABSTRACT

We take into consideration one of the non-linear polynomial combinatorial programming issues known as P-path route minimum distance supply to the destinations from the central station.The distance matrix D(i, j, k) is given from the i th destination to the jth destination utilizing the k th facility, and there are n destinations. A facility, denoted by the letter K, can be an individual component that affects travel distances or costs. Depending on the needs of the many destinations, the central station has the potential to supply the load. The challenge is to determine the lowest distance, while taking the aforementioned factors into account, to connect all of the destinations to the central station (let's say one). We created a Pattern Recognition Technique based Lexi Search Method for this issue, and we built the suggested algorithm in C. We compared it to the models already in use and came to the conclusion that it was recommended for handling higher dimensional issues.

Keywords: *Lexi search method, Pattern recognition technique, Facility*

1. INTRODUCTION

Recently, there has been a lot of focus on the relevance of network advancements in the domain of computing and telecommunications. The design method aims, in part, to achieve maximum connectivity with minimal effort and expense. Some routes contain all of the connections (say P). Planning a road network or an integrated circuit has similar challenges. By imposing limits on the possible degrees a node might have, we can simulate the technological limitation that the number of connections at a node is limited. As Garey [2] shown, the resulting degree-constrained minimum is the lowest possible value. Here, we look into a subclass of Minimum spanning networks. We give computing results for a Lexi-algorithm we created using the "Pattern Recognition Technique" to handle the problem's elementary combinatorial structure.

Minimum Spanning Tree (MST) variants were the focus of the research of a few of the team members. In addition to Pop,et al, [6] and Karger [3], who discovered a randomized linear-time algorithm based on a hybrid of Boruvka's and the reverse-delete method, there are a few others worth mentioning. Chazelle [1] provides a deterministic linear solution to the problem. When the function grows extremely slowly, its duration is o (m, (m,n)). So, Chazelle's algorithm is nearly linear in execution time. Seth Pette [4,5] has discovered a minimum spanning tree methodology that relies on deterministic comparisons and is thus likely optimal. Different spanning models were investigated by Sobhan Babu [8] using a pattern recognition approach [9]. Let there be N nodes that need to reach out to the hub in destination 1. A facility K represents an individual factor that affects the distances and costs. If two destinations, j1 and j2, are linked to i1, then k1 and k2 must be identical. Taking the opposite approach of [11], Suresh Babu [10] investigated a different kind of spanning model. The goal is to find the optimal solution for the minimum distance/cost using k facilities between all destinations connected to headquarters 1.

Problem Description

In this manuscript, we examine the " higher dimensional p – path route minimum distance supply to the destinations from the central station" problem. From the central station "1," the P-trucks (say p = 3) deliver supplies to the other destinations, and the goal is to do so at the lowest possible distance. Each path cannot need more than 100 units of requirement, and there must be at least P destinations connected to the central station via P paths. Central station can accommodate a total of 300 units capacity. To illustrate, assume that N=[1, 2, 3, 4,..... n] destinations have been given distances in the form [NXN]. Using a pattern recognition based lexical search approach, we solve the minimum spanning network connectivity problem in this manuscript. One of the requirements is that all N1 destinations use the same facility. For this manuscript, a precise algorithm is proposed. When applied to the problem of minimizing total travel distance and expenses, the algorithm provides a solution.

Let N denote the set of n stations with the parameters N=1,2,3,4,.....n and K denote the set of k facilities with the parameters 1,2,..., q. If i, j Type equation here.N and k K, then we may write the cost to go from destination i to destination j utilizing facility k as D (i, j, k). Let M=[1, 2, 3...m] be the cluster, and let M be a subset of N if and only if M contains at least m destinations. To illustrate, consider destination 1 to be the central station. The goal is to use P-paths to link the central station to the other (n-1) destinations. Every destination had direct or indirect ties to the central station 1. The P-trucks depart from P-paths in central station1. The number of available P-trucks exceeds or meets the minimum demand of the destinations. The goal of this challenge is to determine the shortest route between all n destinations and the central station, 1. A appropriate numerical example for three different routes demonstrates our Lexi-Search algorithm, which is based on the pattern recognition technique.

Mathematical Formulation

$$\boldsymbol{Minimize\ Z(X)} = \sum_{i\in N}\sum_{j\in N}\sum_{k\in K} D(i,j,k)X(i,j,k) \quad \text{... (1)}$$

Subject to the constraints

$$\sum_{k\in K}\sum_{j=2}^{n} X(1,j,k) = p(=3) \quad \text{... (2)} \qquad \sum_{i\in N}\sum_{j\in N}\sum_{k\in K} X(i,j,k) = n-1$$

$$I(3)$$

Let $\alpha_{r1}, \alpha_{r2}, \alpha_{r3}, \ldots\ldots \alpha_{rn\,r}$ are n_r destinations in r^{th} path(r=1,2,…P) then

$$\sum_{s=1}^{nr-1} x(\alpha_{r\,s,}\ \alpha_{r\,s+1)}) = n_r - 1 \quad \ldots\ (4)$$

Let $\beta_1, \beta_2 \in M$

If $X(\alpha_1, \beta_1, \gamma_1) = X(\alpha_2, \beta_2, \gamma_2) = 1$, Then

$\gamma_1 = \gamma_2$ --- (5) $\sum_{i=1}^{p} n_i = n - 1$ ... (6),

$\sum_{s=1}^{nr} Q(\alpha_{rs}) \le V$ (r=1, 2…P) --- (7)

$$\sum_{i\in N} Q(i) \le PV$$

Where Q (1) =0 … (8)

X (i, j, k) = 0 or 1 -- (9)

Numerical Illustration

To give a concrete illustration of the theory and approach established, let's assume that out of a total of N=1,2,3,4,5,6,7,8,9 destinations, we'll designate destination 1 as the central station and group destinations 2,4,9 into a single cluster (M=2,4,9). It is easy to demonstrate that this is not a required condition when we take the following example in which D(i, j, k) are all non-negative integers. Distance to link Destination 4 with Destination 3 via Facility 1 is shown to be 13 in Table 1. If two destinations are separated by a hyphen ('-'), it means they are not linked in any way. The following table illustrates the destinations' needs. Direct or indirect P-paths link all of the other destinations to the central station. The associated distance matrix is listed below.

Table-1

D(i,j,1)=

-	1	∞	2	7	10	∞	15	18
∞	-	21	∞	25	∞	28	∞	27
∞	23	-	13	28	∞	24	35	40
∞	4	30	-	∞	20	∞	31	2
∞	∞	∞	7	-	22	26	50	29
∞	20	7	5	25	-	10	16	∞
∞	∞	10	∞	20	16	-	12	17
∞	∞	15	6	28	∞	23	-	24
∞	28	12	∞	20	14	17	22	-

Table-2

D(i,j,2) =

-	14	3	22	∞	19	∞	∞	∞
∞	-	20	12	3	12	26	4	32
∞	11	-	18	15	6	4	7	19
∞	∞	29	-	∞	28	∞	18	9
∞	30	∞	∞	-	22	3	24	∞
∞	18	16	16	26	-	∞	24	22
∞	16	15	∞	20	16	-	14	∞
∞	13	∞	30	13	6	18	-	19
∞	24	8	16	∞	26	∞	4	-

We simplify matters by imagining that all destination data is stored in a B-tree of only two dimensions. Then, consider the array B that we provide below:

Table-3

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

Table 2's numerical example shows that destination I is part of cluster M if and only if B (I ss)= 1. However, if B(i) = 0, then the condition is false. Assume that destination 2 is part of M-based cluster if and only if B (2)=1. Take into account the following matrix Q as table-3, which contains the needs of the destinations.

Table-4

1	2	3	4	5	6	7	8	9
-	30	30	20	30	60	40	30	40

From the above table-4, QR (j) = α means that the requirement of destination j is α. Suppose Q(6) = 60 means that the requirement of destination 6 is 60. Here QR(1)= '-' means that the destination 1 act as central station so no need to requirement at 1 but it has the availability 300.

Concepts and Definitions

Definition of a pattern

In this context, the term "pattern" refers to a three-dimensional array of indicators that is linked to a certain task. If Pattern X has a solution (X is a solution), then Pattern is feasible. Specifically: V (x) = D I j, k). An expression of the form X(i, j, k), where i>N, j>N, and k>K. The layout shown in table 4 is a workable one. The cost of connecting the destinations in X is given by the function V(X). Because of this, the entire cost represented by the viable pattern is equal to its value. The method is built in the next chapter, and it looks for the cheapest possible pattern. Sets of ordered triples I (j), (k)] for which X(i), (j), and (k)=1 are representative of patterns in the solution X, with the other X(i), (j), and (k) values being zeros.

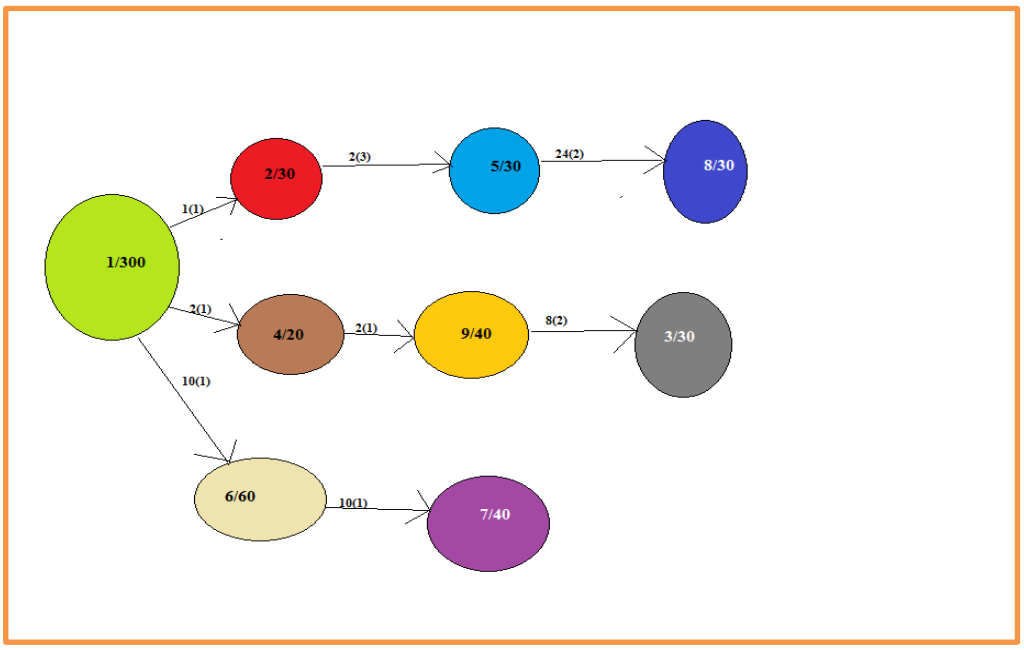
Feasible Solution

Consider an ordered triple set{(1,2,2),(2,5,1),(5,8,1),(1,4,2),(4,9,2),(9,3,1),(1,6,1),(6,7,1)}Represents the pattern given in the table-5, which is a feasible solution.

Table- 5

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

In this context, the term "pattern" refers to a three-dimensional array of indicators that is linked to a certain task. If Pattern X has a solution (X is a solution), then Pattern is feasible. Specifically: V (x) = D (i, j, k). An expression of the form X(i, j, k), where i>N, j>N, and k>K. The layout shown in table 4 is a workable one. The cost of connecting the destinations in X is given by the function V(X). Because of this, the entire cost represented by the viable pattern is equal to its value. Sets of ordered triples I (j), (k)] for which X(i), (j), and (k)=1 are representative of patterns in the solution X, with the other X(i), (j), and (k) values being zeros.



From the above figure, three trucks are started from central station destination 1 with 100 units of capacity.. In first path the destination 1 is connected to destination 2 at facility 1 with 30 units, destination 2 is connected to destination 5 at facility 1, with 30 units and destination 5 is connected to destination 8 at facility 2, with 30 units. In second path destination 1is connected to destination 4 at facility 1 with 20 units, destination 4 is connected to destination 9 with facility1 with 40 units; destination 9 is connected to destination 3 at facility 2 with 30 units. In third path destination 1is connected to destination 6 at facility 1 with 60 units, destination 6 is connected to destination 7 at facility1 with 40 units. So that all the destinations in cluster M (2, 4,9) using the same facility 1 to connect from other destinations. Hence the solution is

Z= D(1,2,1)+D(2,5,2)+D(5,8,2)+D(1,4,1)+D(4,9,1)+D(9,3,2)+D(1,6,1)+D(6,7,1)=1+2+24+2+2 +8+10+10= 59.

Alphabet Table:-

Hihger-dimensional array X contains minimum spanning ordered triples. For our use, they have been indexed from 1 to Z in ascending order of their corresponding distance. Let Z be the number of indices, and let S.No.=[1, 2, 3... Z]. Let D represent the associated cost matrix. To put it another way, if a and b are both S.No. and an is smaller than b, then D (a) is smaller than D (b). Let CD be an array containing the cumulative sum of the components of D, R, C, and F stand in for the row, column, and facility indices of the ordered triples represented by S.No., respectively. Table S-6 lists the numerical example's arrays SN, D, CD, R, C, and F If pSN, then the ordered triple is (R(p),C(p),F(p)) and the value of the ordered triple is D(a)=D(R(a),C(a),F(a)).

Table- 6

S.No.	D	CD	R	C	K
1	1	1	1	2	1
2	2	3	1	4	1
3	2	5	4	9	1
4	3	8	1	3	2
5	3	11	2	5	2
6	3	14	5	7	2

7	4	18	4	2	1
8	4	22	2	8	2
9	4	26	3	6	2
10	4	30	9	8	2
11	5	35	6	4	1
12	6	41	8	4	1
13	6	47	3	6	2
14	6	53	8	6	2
15	7	60	1	5	1
16	7	67	5	4	1
17	7	74	6	3	1
18	7	81	3	8	2
19	8	89	9	3	2
20	9	98	4	9	2
21	10	108	1	6	1
22	10	118	6	7	1
23	10	128	7	3	1
24	11	139	3	2	2
25	12	151	7	8	1
26	12	163	9	3	1
27	12	175	2	4	2
28	12	187	2	6	2
29	13	200	3	4	1
30	13	213	8	2	2
31	13	226	8	5	2
32	14	240	9	6	1
33	14	254	1	2	2
34	14	268	7	8	2
35	15	283	1	8	1
36	15	298	8	3	1
37	15	313	3	5	2
38	15	328	7	3	2
39	16	344	6	8	1
40	16	360	7	6	1
41	16	376	6	3	2

42	16	392	7	2	2
43	16	408	7	6	2
44	16	424	9	4	2
45	17	441	7	9	1
46	17	458	9	7	1
47	18	476	1	9	1
48	18	494	3	4	2
49	18	512	4	8	2
50	18	530	6	2	2
51	18	548	8	7	2
52	19	567	1	6	2
53	19	586	3	9	2
54	19	605	6	5	2
55	19	624	8	9	2
56	20	644	4	6	1
57	20	664	6	2	1
58	20	684	7	5	1
59	20	704	9	6	1
60	20	724	2	3	2
61	20	744	7	5	2
62	21	765	2	3	1
63	22	787	5	6	1
64	22	809	7	9	1
65	22	831	1	4	2
66	22	853	5	6	2
67	23	876	3	2	1
68	23	899	8	7	1
69	24	923	3	7	1
70	24	947	8	9	1
71	24	971	5	8	2
72	24	995	6	8	2
73	24	1019	9	2	2
74	25	1044	2	5	1
75	25	1069	6	5	1
76	26	1095	5	7	1

77	26	1121	2	7	2
78	26	1147	6	4	2
79	26	1173	9	6	2
80	27	1200	2	9	1
81	28	1228	2	7	1
82	28	1256	3	5	1
83	28	1284	8	5	1
84	28	1312	9	2	1
85	28	1340	4	6	2
86	29	1369	5	9	1
87	29	1398	4	3	2
88	30	1428	4	3	1
89	30	1458	5	2	2
90	30	1488	8	4	2
91	31	1519	4	8	1
92	32	1551	2	9	2
93	35	1586	3	8	1
94	40	1626	3	9	2

Let us consider 5□S.No. It represents the ordered triple (R(5),C(5),(5)) =(2,5,2).Then D(5) = 3and CD

(5) =11.

4.4 Definition of an Alphabet – Table and a word

D is an array of the corresponding costs of the ordered triples, and DC is an array of the cumulative sums of components in D, where SN = (1,2,...) is the set of indices. Let R, C, and k represent the row, column, and facility indices of the arrays containing the sorted triples, respectively. Let the series of k indices from SN be denoted by Lk = a1, a2,...., ak, where ai is an index from SN. The sequence order ai has no bearing on the pattern depicted by the ordered triples whose indices are given by Lk. Therefore, the indices are 886rganized in a way that guarantees uniqueness: ai ai+1, where I = 1, 2,...., k-1. In this context, the ordered sequence Lk is a "word" of length k, and the set SN is the "Alphabet-Table," where the indices range from 1 to n2p. A "sensible word" Lk is one that makes sense. An insensible word exists if and only if (for all I from 1 to k-1) ai is less than ai+1. If the related pattern X is feasible, then the word Lk is said to be feasible. The same holds true for infeasible and partially feasible patterns. If the block of words that Lk represents contains at least one feasible word, or if the partial pattern that Lk represents does not contain any inconsistencies, then we say that Lk is a viable partial word.

The initial position of the letters SN in the unfinished word Lk can be any of the letters in SN. Any conceivable pattern can only have n-1 unit entries, hence we're only interested in the set of words up to and including n-1 characters in length. Whenever k is less than n, Lk is referred to as a short word, and whenever k equals n, it is referred to as a long word. An Lk-representing word fragment is a set of words that follows Lk as its leader letters (the first k). If the set of words that the leader specifies contains at least one possible word, we say that the leader is feasible.

4.5 Algorithm- : (Lexi – Search algorithm)

STEP 0: (Initialization)

The arrays SN, D, DC, R, C, T and LN the values of N, M are made available IR,IC,SW,SWI,F, IK, L, V, LB are initialized to zero. The values I=1, J=0, LN (TR) =0, TV= □ , MAX= (n*n)/2

STEP I:	J=J+1 MAX= (n*n)/2LDP=0 IS (J>MAX)	IF NO GOTO II IF YES GOTO XXX
STEPII:	L (I) =J	
	TR= (J)TC= C (J)TK= T (J)	GOTO III
STEPIII:	V (I) =V (I-1) +D (J) LB (I) =V (I) +CD (J+N-1-I)-CD (J))	GOTO IV
STEPIV:	IS (LB (I)>=VT)	IF NO GOTO V IF YES GOTO XXXIV
STEPV:	IS (TR= =HC)	IF YES GOTO VI
STEPVI:	IS (IR [TR] <P)	IF NO GOTO VII IF YES GOTO VIII
STEPVII:	IS (IR [TR] = =1)	IF NO GOTO II IF YES GOTO II
STEPVIII:	IS(IC [TC] = =1)	IF NO GOTO VIII IF YES GO TO II
STEP*IX*:	W=TC	IF NO GOTO IX GOTO X

STEPX:	LDP=LDP + q[w] IS (LD<LDP)	IF NO GOTO XII IF YES GOTOII
STEPXII:	IS (SW[w] = =0)	IF NO GOTO XIII IF YES GOTOXV
STEPXIII:	IS (W= =TR))	IF NO GOTO XIV IF YES GOTOII
STEPXIV:	W=SW [W]	GOTOX
STEPXV:	W=TR	GOTOXVI
STEPXIV: STEPXVII:	LDP=LDP + q[w]IS (LD<LDP) IS (SWI[w] = =0)	IF NO GOTO XVII IF YES GOTOII IF NO GOTO XVIII IF YES GOTOXX
STEPXVIII:	IS (W= =TC))	IF NO GOTO XIX IF YES GOTOII

STEPXVIX:	W=SWI [W]	GOTOXVI
STEPXX:	IS (b [TC] = =1)	IF NO GOTO XXV IF YES GOTOXXI
STEPXXI: STEPXXII:	IS (NB= =0) B=b [TC]	IF NO GOTO XXIII IF YES GOTOXXII
	F [B] = TK	GOTOXXIV
STEPXXIII:	IS (F [B] = = TK)	IF NO GOTO II IF YES GOTOXXIV
STEPXXIV:	NB= NB+1	GOTO XXV
STEPXXV: STEPXXVI:	IS (i= =n-1) L [I] =J	IF NO GOTO XXVI IF YES GOTOXXVIII
	IC [TC] =1 SW [TR] =TC SWI [TC] =TRIS (TR= =HC)	{IR [TR] =IR [TR] +1 GOTO XXVII}
STEPXXVII:	I=I+1	IR [TR] =1 GOTO XXVII}
	Z2=P-IR [HC] IS (Z2<=n-I)	IF YES GOTOII
STEPXXVIII:	T=V [I]	IF NO GOTO XXIX
	L [I] =J	GOTO XXIX
STEPXXIX:	I=I-1	GOTO XXX
STEPXXX:	J=L [I] TR=R [J]TC=C [J]	

	IR [TR] =0 IC [TC] =0 SW [TR] =0 SWI [TC] =0	
	L [I+1] =0 IS (TR= =HC)	IR [HC] = IR [HC]-1 GOTO XXXI}
STEP XXXI:	IS (b [TC] = =1)	IR [TR] =0 GOTO XXXI} IF YES GOTOXXXII
STEP XXXII:	NB=NB-1	IF NO GOTO 2 GOTO XXXIII

STEPXXXIII:	IS (I= =1)	IF YES GOTO XXXIV IF NO GOTO XXIX
STEP XXXIV:	STOP	

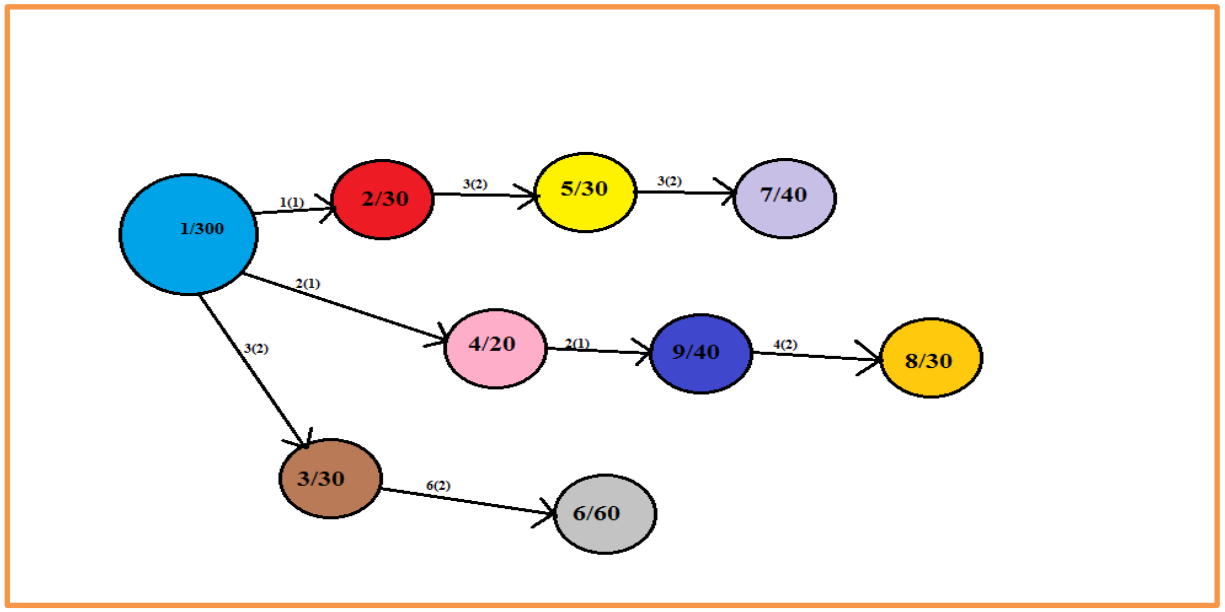
4.6 Search-Table:

The working details of getting an optimal word using the above algorithm for the illustrative numerical example is given in the Table-7. The columns named (1), (2), (3),…, gives the letters in the first, second, third and so on places respectively. The columns R, C and K give 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 -7, Y indicates ACCEPT, N indicates REJECT and TV is the feasible solution or optimal solution..

TABLE-7

SN	1	2	3	4	5	6	7	8	V	LB	R	C	K	REMARK
1	1								1	22	1	2	1	Y,(F=1)
2		2							3	22	1	4	1	Y (F=1)
3			3						5	22	4	9	1	Y,(F=1)
4				4					8	22	1	3	2	Y
5					5				11	22	2	5	2	Y
6						6			14	22	5	7	2	Y
7							7		18	22	$4^{\#}$	2	1	N
8							8		18	22	$2^{\#}$	8	2	N
9							9		18	22	3	6	2	Y
10								10	24	24	9	8	2	Y,TV=24
11							10		18	24	9	8	2	N,≥TV
12						7			15	24	$4^{\#}$	2	1	N, ,≥TV
13					6				11	24	5	7	2	N, ,≥TV
14				5					8	24	2	5	2	N, ,≥TV
15			4						6	25	1	3	2	N, ,≥TV
16		3							3	26	4	9	1	N, ,≥TV
17	2								2	24	1	4	1	N

At the end of the search the current value of TV is 23 and it is the value of the optimal feasible word $L_{8=}$=(1, 2, 3, 4, 5, 6, 7, 9,10). It is given in the 10^{th} row of the search table. The optimal feasible word $L_{8\ =}$ (1, 2, 3, 4, 5, 6, 7, 9,10) and the respective ordered triple set is{(1,2,1),(1,4,1),(4,9,1),(1,3,2),(2,5,2),(5,7,2),(3,6,2)(9,8,2)}.The paths represented by the above pattern is {(1,2,1),(1,4,1),(4,9,1),(1,3,2),(2,5,2),(5,7,2),(3,6,2)(9,8,2)}The destinations {2, 4, 9} using the same facility 1. The diagrammatic representation of this solution can also see in the following figure-1.



The above figure-1 represents a optimal solution. In first path the central station 1 is connected to the destination 2 at facility 1 with 30units, destination 2 is connected to destination 5 at facility 2, with 30 units , destination 5 is connected to destination 7 at facility 2, with 40 units and the sum of the total supply to the destinations in first path is 100 units, which is less than or equal to truck capasity100. In second path, the central station 1 is connected to destination 4 at facility 1 with 20 units, destination 4 is connected to destination 9 with facility 1 with 40 units, and destination 9 is connected to destination 8 at facility 2 with 30 units and the sum of the total supply to the destinations in second path is 90 units(≤100), which is less than or equal to the truck capasity100. In third path central station is connected to destination 2 at facility 1 with 30 units, destination 2 is connected to destination 5 at facility2 with 30 units; destination 5 is connected to destination 7 at facility 2 with 40 units and the sum of the total supply to the destinations in first path is 80 units, which is equal to the truck capasity100. So, in each path sum of the supply to the destinations less than or equal to the truck capasity100 and all the destinations in cluster M (2, 4, 9) using the same facility 1 to connect from other destinations Hence the optimal solution is Z= D(1,2,1)+ D(2,5,2)+ D(5,7,2)+D(1,4,1)+ D(4,9,1)+ D(9,8,2)+D(1,3,2)+ D(3,6,2) =1+3+3+2+2+4+3+6= 24

2. 5 EXPERIMENTAL RESULTS

Results obtained for the suggested algorithm, a pattern recognition tool built on the Lexi-search algorithm, are shown in the table below. We implement this technique as a C programme and test it on a COMPAQ dx2280 MT computer. Through testing with a range of input sizes, we validate this algorithm's effectiveness. We populate the distance matrix with arbitrary numbers. D I j, k) in the distance matrix is assigned uniformly random numbers between 0 and 1000. Changing the parameters N, K, and CL allowed us to test a variety of scenarios. A tabular presentation of the findings (Table -10) is provided below. Seven to nine data sets are examined in each case. It can be observed that the time needed to find the best option is relatively short.

Table-10

SN	Problem dimension			TPN	CPU runtime in seconds	
	N	K	LC		Avg TA	Avg ST
1	4	2	1	7	0	0
2	4	2	2	7	0	0
3	5	2	1	7	0	0
4	5	3	1	7	0.054945	0
5	10	2	1	7	0.109890	0
6	10	3	1	7	0.21978	0

7	10	3	2	7	0.219780	0
8	15	2	1	9	0.384615	
9	15	3	1	9	0.659341	0.054945
10	15	3	2	9	0.329670	0.054945

In the above table the notation N represent number of destinations, K represent number of facilities, LC represent number of clusters to be taken, TPN represents number of problems tried. In the next columns Avg. TA represent average central processing unit run time to form an alphabet table. Avg. TS represents average central processing unit run time to form search table for obtaining the optimal solution. We observe that the time requirement of search table very less comparatively of time duration of alphabet table.

3. CONCLUSION

In this manuscript, we looked at the model " higher dimentional p - path route minimum distance supply to the destinations from the central station" and created a new algorithm based on a pattern recognition technique to find the optimal solution. Then, the model is called as a zero-one programming issue. Each and every step of the new algorithm and related concepts are discussed, along with a relevant numerical example. To implement the our proposed algorithm, we used the C programming language.. Since higher dimensional problems require less CPU time to run, the best possible answer can be found using this method. In addition to that, Lexi search techniques have been shown to be more effective in a wide variety of combinatorial issues. In numerous studies, researchers have demonstrated the effectiveness and speed of their own Lexi - search algorithms by employing various alphabet tables. From our observations, we conclude that this approach is quite efficient and can handle problems of much greater sizes.

REFERENCES

[1] Garey, Michael R., Computers and Intractability, A Guide to the Theory of NP- Completeness, 1979, ISBN 0-7167-1045-5.

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

[3] P. Seth, R. Vijaya, A randomized time-work optimal parallel algorithm for finding a minimum spanning forest, SIAM Journal on Computing 31, (2002) 1879{1895, doi:10.1137/S0097539700371065.

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

[5] Reeves, C.R., "Moderen Metaheuristics Techniques for Combinatorial Problems", Blackwell, Oxford, 1993.

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

[7] C Suresh Babu, K Sobhan Babu and M Sundara Murthy, Variant Minimum Spanning Network Connectivity Problem,International Journal of Engineering Science and Technology,3,2011,

[8] P. Biswas, M Goel, H Negi and M Datta, An Efficient Greedy Minimum Spanning Tree Algorithm Based on Vertex Associative Cycle Detection Method, Procedia Computer Science, 92(2016),513 {519, doi:10.1016/j.procs.2016.07.376.

[9] P. Seth & R. Vijaya.,An Optimal Minimum Spanning Tree Algorithm, Journal of the ACM,49, (2000), 49{60, doi:10.1007/3-540-45022-X_6.

[10] P. Ayegba, J. Ayoola, E. Asani and A. Okeyinka, A Comparative Study Of Minimal Spanning Tree Algorithms, International Conference in Mathematics, Computer Engineering and Computer Science,2020, 1{4, doi: 10.1109/ICMCECS47690.2020.240900.

[11] Osaba, Eneko & Yang, Xin-She & Del Ser, Javier. (2020). Traveling salesman problem: a perspective review of recent research and new results with bio-inspired metaheuristics. 10.1016/B978-0-12-819714-1.00020-8.

[12] C Hansknecht, I Joormann and S.Stiler, Dynamic Shortest Paths Methods for the Time-Dependent TSP, Algorithms, 14,202,,1{3,doi:10.3390/a14010021.