International Journal of Scientific and Innovative Mathematical Research (IJSIMR)
Volume 3, Issue 4, April 2015, PP 21-33
ISSN 2347-307X (Print) & ISSN 2347-3142 (Online)
www.arcjournals.org

Generalized Travelling Salesman Problem with Clusters

Guravaraju Pokala

Research Scholar, Dept of Mathematics
S.V. University, Tirupati-517501
guravarajup@gmail.com

Sreenuvasulu Yattapu

Research Scholar, Dept of Mathematics
S. V.University, Tirupati-517501
srinivas.srist@gmail.com

Prof.M.Sundara Murthy (Rtd)

Dept of Mathematics, S.V.University
Tirupati-517501
profmurthy@gmail.com

Abstract: *The Travelling salesman problem(TSP) is a popular Combinatorial Programming Problem. Many times for TSP model the the basic feature is that the salesman starts at the headquarter city and visits each of the cities once and only once and returns to the headquarter.*

In this problem a variant Travelling salesman problem called ***"GENERALIZED TRAVELLING SALESMAN PROBLEM WITH CLUSTERS"*** *states that,* ***N*** *be the set of* ***n*** *cities,* ***N={1,2,3,……n}*** *and here the city "1" is taken as head quarter city. M represents a cluster which is a subset of N. the revisiting city is represented by* ***α****. The distances between cities are represented by matrix D. The salesman starts his tour from Head quarter city "1" visit some cities and reaches city* ***α****, deviates from that city tours some cities in the cluster and revisits city* ***α*** *and continues his tour by visiting other cities and returns head quarter using different facilities . When* ***'r'*** *clusters of cities are there we can think of a maximum of* ***r*** *revisits in the tour. For this tour we calculate the total cost/distance which includes the revisiting city {****α****}. Among several tours of salesman with the above condition, we want that tour for which total distance/coast is minimum.*

In this sequel, we develop an Alphabet table and, search table to find feasible tour and optimal tour. For this we develop Lexi -search algorithm using pattern recognition technique.

Keywords: *Travelling salesman, cluster, Headquarter, Alphabet table, Search table, Lexi-search algorithm, Optimal solution, Feasible solution*

1. Introduction

The traveling salesman problem(TSP) is one of the most popularly studied combinatorial programming problem in the Operations Research literature. There are so many researchers have been developed different algorithms for the solution of TSP so far. It is a kind of mathematical puzzle with a long enough history.

Suppose a salesman wants to visit a certain number of cites allotted to him. He knows the distance / cost of every pair of cites i and j denoted as D. The problem is to select a route that starts from a given home city (head quarter) to passes through each and every city once and only once and returns to his starting city (head quarter) in the shortest distance. Here the objective of the problem is to find a "Tour" in such a way that with minimum distance / cost. Here, in the present study we have considered a variation of the above traveling salesman problem.

2. Variation of Travelling Salesman Problem

There are so many researchers have been developed different algorithms for the solution of TSP so far. But the problem has not received much attention in its restricted context. However, literature which is available with regard to the TSP with variations are discussed Das[2], Kubo & Kasugai[1], Pandit[3], Ramesh[4], Raviganesh et.all.[5] and Srivastava Kumar et al[6].

The visit of city (cities) in TSP has certain significance. It is observed that sometimes revisit of city (cities) may be cheaper than when revisit is not allowed. Revisit of city may be either 'a must' (due to lack of communication) or more economical. It has been shown in (Das-1971) that the example discussed in Little, et al.[7], problem yields cheaper route when revisit is allowed at subset of cities (cluster). Sundara Murthy,M(1979) [8] Combinatorial Programming-A Pattern Recognition Approach Hardgrave,W.W & G.L.Nambhauser(1962)[9]:On the Relation between the Travelling Salesman and the Longest Path Problems and Flood, M.M.(1956)[10]:The TSP Operations Research,

3. Generalized Travelling Salesman Problem (GTSP)

The GTSP finds practical application particularly in many variants of routing problems e.g., when some good can be delivered to multiple alternative addresses of customers. There exist several applications of the GTSP such as 1) Postal routing, 2) Computer file processing, 3) Order picking in ware houses, 4) Process planning for rotational parts and 5) The routing of clients through welfare. Occasionally, such application can be directly modeled as the GTSP, but more often the GTSP appears as a sub problem. Furthermore, many other combinatorial optimization problems can be reduced to the GTSP problems. The GTSP is NP-hard since it is a special case of the TSP which is partitioned into m clusters with each containing only one node.

4. Problem Description

Number of researchers studied this problem with many constraints and generalizations. Some studied with three dimensional distance matrix.

Among the available cities if a sub set of cities are in a cluster then it may be convenient for the salesman to deviate from a suitable city and visit the cities in the cluster once and only once and revisit the deviating city and continuing the tour. If this revisiting is permitted, then the total distance travelled in his tour may be lessthan a tour where the revisit is not permitted. So whenever clusters of cities are there this type of possibility can be thought of. Depending on the number of clusters we can permit suitable number of revisiting of cities. That way we can claim this TSP model as one type of generalization.

Let there be n cities where N={1,2,3,…….,n}.Let M be the sub set of n cities where $M \subset N$ and m<n.Let m = IMI be the cluster of cities in the set N($M\subset N$) Let 1 be the Head quarter city and $\alpha \in M$ be the city where the sales man revisits the cities in his tour. Then let d(i,j,k) be the cost/distance of salesman visiting from i^{th} city to city j with using facility k. Generally 'k' is a special factor (or) independent factor which influences the cost from city i to city j.

The cost matrix D is given. The salesman should starts his tour from Head quarter city 1 visits some cities and when he reaches city $\alpha \in M$ deviates from that city tours m cities in the cluster city and revisits city α and continues his tour by visiting other cities and returns Head quarter city 1.For the tour we can calculate the total cost which includes the revisiting city of the α. Among the several tours of the sales man with the above condition, we want that tour for which the total cost is minimum.

So in this sequel, we will develop a Lexi search algorithm using pattern recognition technique and find the optimal tour. When 'r' clusters of subset cities are there we can think of a maximum of r revisits of cities in the tour. For this model also our algorithm gives an optimal tour. If there is a possibility of a tour for the salesman without revisit of the cities with least cost, that solution also can be made available in this algorithm.

5. Mathematical Formulation

$$\textbf{Min } Z = \sum_{i\in N}\sum_{j\in N}\sum_{k\in K} D\,(i,j,k)\,X\,(i,j,k) \qquad (1)$$

where N ={1,2,3…n}, K= (1,2,…r) , $M_i = (a_i, a_2, a_3 \ldots a_{mi})$, i=1,2….n, $Mi \subset N$

Subject to the conditions

$$\sum_{i\in N}\sum_{j\in N}\sum_{k\in K} X\,(i,j,k) \le n + r \qquad (2)$$

There is tour in M_i cities of the cluster i^{th} cluster with **(3)**

α_i as revisiting city $i=1,2 \ldots\ldots n$

If $X(i_1, j_1, k_1) = X(j_1,j_2,k_2)=1$ **(4)**

$J_1 \in N$ then $j_1 \neq 1$, $k_1 \neq k_2$

$I_1,j_2 \in N$ and $k_1,k_2 \in K$

$X(i,j,k)= 0$ (or) 1 , $i,j \in N$, $k \in K$ **(5)**

The eqn.(1) represents the objective of the problem i.e., to find total minimum distance / cost to connect from all the cities to the Headquarter city.

The eqn. (2) represents the total number of arcs in the salesman tour.

The eqn.(3) represents a tour in the cluster city M_i starting from the revisiting city α_i and return to the same city.

The eqn.(4) represents the salesman starts tour from $i_1 \longrightarrow j_1$ and $j_1 \longrightarrow j_2$ if he travelling, he should use different facilities. i.e. $k_1 \neq k_2$ and it is not true when $j_1=1$ city.

The eqn. (5) represents if the i^{th} city is connecting j^{th} city, it is 1 otherwise Zero

6. Numerical Illustration

The algorithm and concepts are developed for a numerical example. For this the total no. of cities are taken as n=6 i.e N = {1, 2, 3, 4, 5, 6},k=(1,2) The distance D(i, j,k) is represented as D(i,j,1) and D(i,j,2) in the following Tables 1 &2 Using the Tables we find the optimum solution in Lexi-search approach using the "pattern recognition technique". Here city 1is taken as Headquarter. The salesman starts from Headquarter city (i.e., city 1), visits all the remaining cities and come back to Headquarter city by using facilities k. In this tour the salesman visits in a Cluster and revisits one city in that cluster.

In the table the distance (i, i) is taken as ∞ as they are not involved in the tour. Here the entries D taken as non-negative integers, it can be easily seen that this is not a necessary condition.

The distance matrices is given in the following Tables.

Table -1

D(i, j,1)

∞	7	26	10	3	23
14	∞	3	21	17	12
2	11	∞	5	8	19
20	1	13	∞	15	22
9	18	27	16	∞	2
3	25	6	10	24	∞

Table-2

D(i,j,2)

∞	2	22	9	17	8
25	∞	14	1	4	16
13	15	∞	21	10	21
27	4	18	∞	24	12
7	26	23	3	∞	5
28	19	1	6	11	∞

From the above table-1, D (4,2,1)=1 means the distance/cost of the connecting the city 4 to city 2 is 1 units using facility1. The cluster is $M=(2,4,5) \subset N$ and i=1is one cluster only.

7. Feasible Solution

Consider the ordered triples (4,2,1), (2,4,2), (6,3,2), (3,1,1), (5,6,1), (2,5,2), (1,2,1) represents a feasible solution. The object is to find a tour starting from '1'for the 5 cities with minimum total distance with the condition of revisiting of a city in the cluster M=(2,4,5).

In the following Figure-1, the rectangle represents headquarter, circle represents revisiting city and triangles represents cities, the values in above geometrical figures indicates name/number of the cities. Also value on each arc before parenthesis represents distance between the respective two nodes and values in parenthesis represent the facility used.

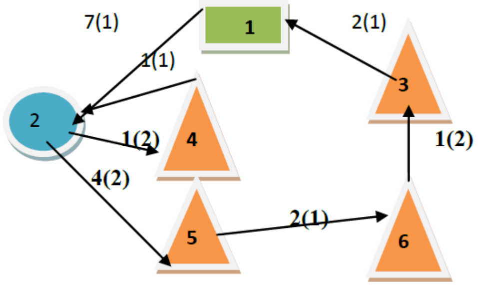

Figure1

From the above figure-1, the salesman started his tour from city 1 (headquarter) and visited city 2 using facility 1,city 2 to city 4 with facility 2,city 4 to city 2 with facility 1, again city 2 to city 5 availing facility2, city 5 to city 6 with facility 1, city 6 to city 3 using facility 2 and city 3 to city 1 (head quarter) availing facility 1. From the figure it is clear that city 2 is identified as a revisiting city. The value of the feasible solution is 18.

Z=D(4,2,1)+D(2,4,2)+D(6,3,2)+D(3,1,1)+D(5,6,1)+D(2,5,2)+D(1,2,1)

=1+1+1+2+2+4+7=18

8. Solution Approach

In the above figure-1 for the feasible solution we observe that there are 7 ordered triples taken along with the values for the numerical example in dist/cost tables. The 7 ordered triples are selected such that they represent a feasible solution in figure-1. So the problem is that we have to select 7 ordered triples from the distance matrices along with values such that the total value is minimum/least and represents a feasible solution. For this selection of 7 ordered triples we arrange the 6 x 6 x 2 ordered triples in the increasing order of costs and call this formation as alphabet table and we will develop an algorithm for the selection along with the checking for the feasibility.

9. Concepts and Definitions

9.1. Definition of a Pattern

A indicator three-dimensional array X which is associated with the number of cities connecting is called a ‘pattern’. A pattern is said to be feasible if X is a solution.

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

The pattern represented in the Tables 3 & 4 represents feasible pattern. The value V(X) gives the total distance/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.

Consider an ordered triples set (4,2,1), (2,4,2), (6,3,2), (3,1,1), (5,6,1), (2,5,2) and (1,2,1) represents the pattern given in the Tables-3 & 4, which represents feasible solution in figure 1.

Table-3

D(i,j,1)

$$\begin{bmatrix} 0&1&0&0&0&0 \\ 0&0&0&0&0&0 \\ 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&0&0&0&1 \\ 0&0&0&0&0&0 \end{bmatrix}$$

Table-4

D(i,j,2)

$$\begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&0&1&1&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&1&0&0&0 \end{bmatrix}$$

9.2. Alphabet Table

There are n x n x 2 ordered triples in three-dimensional array D. For convenience these are arranged in ascending order of their corresponding cost/distance and are indexed from 1,2,…..

(Sundara Murty-1979). Let SN=[1, 2, 3….], be a set of indices. Let D be the corresponding array of distances. For our convenience we use the same notation D. If a, b $\in$ SN and a < b then D (a) $\leq$ D(b). Also let the arrays R, C and K be the array of row, column and facility indices of the ordered triples represented by SN and CD be the array of cumulative sum of the elements of D. The arrays SN, D, CD, R, C and K for the numerical example are given in the Table-5. If p $\in$ SN then (R(p), C(p), K(p)) is the ordered triples and D(a) = D(R(a),C(a), K(a)) is the value of the ordered triples and CD(a) $=\sum_{i=1}^{a} D(i)$.

Table5. *Alphabet Table*

S. NO .	D	CD	R	C	K
1	1	1	4	2	1
2	1	2	2	4	2
3	1	3	6	3	2
4	2	5	3	1	1
5	2	7	5	6	1
6	2	9	1	2	2
7	3	12	1	5	1
8	3	15	2	3	1
9	3	18	6	1	1
10	3	21	5	4	2
11	4	25	2	5	2
12	4	29	4	2	2
13	5	34	3	4	1
14	5	39	5	6	2
15	6	45	6	3	1
16	6	51	6	4	2
17	7	58	1	2	1
18	7	65	5	1	2
19	8	73	3	5	1
20	8	81	1	6	2
21	9	90	5	1	1
22	9	99	1	4	2
23	10	109	1	4	1
24	10	119	6	4	1
25	10	129	3	5	2
26	11	140	3	2	1
27	11	151	6	5	2
28	12	163	2	6	1
29	12	175	4	6	2
30	13	188	4	3	1
31	13	201	3	1	2
32	14	215	2	1	1
33	14	229	2	3	2
34	15	244	4	5	1
35	15	259	3	2	2
36	16	275	5	4	1
37	16	291	2	6	2
38	17	308	2	5	1
39	17	325	1	5	2
40	18	343	5	2	1
41	18	361	4	3	2
42	19	380	3	6	1
43	19	399	6	2	2
44	20	419	4	1	1
45	21	440	2	4	1
46	21	461	3	4	2
47	21	482	3	6	2

48	22	504	4	6	1
49	22	526	1	3	2
50	23	549	1	6	1
51	23	572	5	3	2
52	24	596	6	5	1
53	24	620	4	5	2
54	25	645	6	2	1
55	25	670	2	1	2
56	26	696	1	3	1
57	26	722	5	2	2
58	27	749	5	3	1
59	27	776	4	1	2
60	28	804	6	1	2

Let us consider 12 $\in$ SN. It represents the ordered triples (R (12), C (12), K (12)) = (4,2,2). Then D(12)=D(4,2,2) =4 and CD(12)=29

9.3. Definition of a Word

Let SN = {1, 2……} be the set of indices, D be an array of corresponding distances of the ordered triples and cumulative sum of elements of D is represented as an array CD. Let arrays R, C and K be respectively, the row, column and facility indices of the ordered triplets. Let L_k={a_1, a_2, ……, a_k}, $a_i \in$ SN be an ordered sequence of k indices from SN. The pattern represented by the ordered triplets 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 for i=1, 2, ………k-1. The set SN is defined as the "alphabet-Table" with alphabetic order as 1, 2, …, n and the ordered sequence L_k is defined as a "word" of length k. A word L_k is called "sensible word". If $a_i < a_{i+1}$ 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 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.

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

9.4. Value of the Word

The value of the partial word L_k, $V(L_k)$ is defined recursively as $V(L_k) = V(L_{k-1})+D(a_k)$ with $V(L_o)$=0 where D (a_k) is the distance/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. X is the (partial) pattern represented by L_k, (Sundara Murthy – 1979).

For example L_4={1,2,3,4}

$V(L_k) = V(L_{k-1})+D(a_k)$

$V(L_4) = V(L_3)+D(a_4)$

=3+2=5

9.5. Lower Bound of a Partial Word Lb (Lk)

A lower bound LB (L_k) for the values of the block of words represented by

$L_k = a_1, a_2........, a_k$ can be defined as follows.

$LB(L_k) = V(L_k) + CD (a_k + n+1 – k) – CD (a_k)$

$LB(L_4) = V(L_4) + CD (a_4 + 6+1 – 4) – CD (a_4)$

= 5 + CD(4+7-4) – CD(4)

= 5+ CD(7) – CD(4)

= 5+ 12-5=12

LB(L_4) =12

9.6. Feasibility Criterion of a Partial Word

An algorithm was developed, in order to check the feasibility of a partial word $L_{k+1}=(a_1, a_2, \ldots\ldots\ldots a_k, a_{k+1})$ 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 $\in$ N indicates and the i^{th} city is connected to city j. Otherwise IR(i) = 0.
- IC be an array where IC (j) = 1, j $\in$ N indicates and the j^{th} city is connected by city i. Otherwise IC (j) = 0.
- SW be an array where SW (i) = j indicates that the i^{th} city is connected to city j. Otherwise SW (i) = 0.
- L be an array where L(i) = a, i $\in$ N, a_i εSN is the letter in the i^{th} position of word.
- M be an array where M(i) =1, where i εN in the cluster i.e., i^{th} city is connected to city j in cluster otherswise M(i)=O.

The value of the arrays IR, IC,IK, SW, L are as follows

- IR (R(a_i)) = 1, i=1, 2, ……… k and IR (j) = 0 for other elements of j.
- IC (C(a_j)) = 1, j=1, 2, ……… k and IC (i) = 0 for other elements of i.
- SW (R(a_i)) = C (a_i), i = 1, 2, …. k and SW(j) = 0 for other elements of j.
- L(i) = α_i, i = 1, 2, ….., k, and L(j) = 0 for other elements of j.
- IK(i) = 1,i=1,2…k and K(j) =0 for the other elements of j.

For example consider a sensible partial word L_4 = (1, 2, 4, 6) which is feasible. The array L, IR, IC, SW takes the values represented in table-6 given below.

Table6.

	1	**2**	**3**	**4**	**5**	**6**
L	**1**	**2**	**3**	**4**	**-**	**-**
IR	**-**	**1**	**1**	**1**	**-**	**1**
IC	**1**	**1**	**1**	**1**	**-**	**-**
SW	**-**	**4**	**1**	**2**	**-**	**3**
IK	**-**	**2**	**1**	**1**	**-**	**2**

10. ALGORITHM-1 (FEASIBLE CHECKING)

Step 0: IS IX=0 go to101

Step 101: IS (TR=HC) go to 107

Else go to 102

Step102: IS (IR(TR)=2) go to 2

Else go to 103

Step 103: IS (IC(TC)=1 go to 2

Else go to 104B

Step 104 B: Z=P-NP

RP=n-1-i

Is (RP≥7) go to 104

Else go to 2

Step 104: W=TC go to 105

Step 105: IS (SW(w)=0) go to 108

Else go to 106

Step106: IS (W=TR) go to 2

Else go to 7

Step 107 : W=SW(W) go to 105

Step 108: IS IX=1

Step2 :- STOP.

Algorithm2

Step:1. Initialization

The a SN, D, CD,R,C,M,N are made available IR, IC, L, V, LB& SW are initialzed to zero. The values I=1, 5=0

VT=999, MAX=N & N, P=2

STEP:2J =J+1 yes go to 14

I8(S>Max) no go to 3

Step:3. L(I)=5

TR= R(5)

T(C) =C(5) go to 4

Step:4. V(I)=V(I-1)+D(5)

LB(I) =V(I)+CD(J+N-1-I)-CD(5) go to 5

Step : 5. IS(LB (I,$\geq$VT) YES GO TO 16

No go to 6

Step:6. (check the feasibility of using algorithm-1)

IS IX=1 yes go to 7

Else no go to 6

Step7: IS (I=N-1) yes go to 10

Else no go to 8

Step 8: L(I) =5

IR (TR)=1 IR (TR)= IR (TR)+1

SW (TR)=TC yes go to 9

IS (TR=HC) go to 9

Else

Step9: I=I+1

EZ = P-IR(HC)

IS (EZ$\leq$n-1) yes go to 2

Else no go to 14

Step10: IS HC =TC yes go to 11

Else no go to 12

Step11: IS (IR(HC)=P-1) yes go to 13

Else go to 2

Step13: VT=V(I)

L(I)=5 go to 14

Step14: I=I-1

I=L(I)

TR=R(5)

TIC (5)

IR(TR)=0

SW(TR)=0

L(I+1)=0 go to 16

Step 16: IS I=1 yes go to17

Step 17: Stop

11. Search Table

The working details of getting an optimal word using the above algorithms for the illustrative numerical example is given in the Table-6. The columns named (1), (2), (3), (4), (5), (6) & (7) gives the letters in the first, second, third and so on places respectively. The corresponding V and LB are indicated in the next two columns. The columns R, C and K gives the row, column and facility indices of the letter. The last column REM gives the remarks regarding the acceptability of the partial words. In the following table A indicates ACCEPT and R indicates REJECT.

Table7. *Search Table*

SN	1	2	3	4	5	6	7	V	LB	R	C	K	REM
1	1							1	9	4	2	1	A
2		2						2	12	2	4	2	A
3			3					3	12	6	3	2	A
4				4				5	12	3	1	1	A
5					5			7	12	5	6	1	A
6						6		9	12	1	2	2	R
7						7		10	13	1*	5	1	R
8						8		10	13	2	3*	1	R
9						9		10	13	6	1*	1	R
10						10		10	14	5*	4	2	R
11						11		11	15	2	5	2	A
12							12	15	15	4	2	2*	R
13							13	16	16	3*	4	1	R
14							14	16	16	5	6	2*	R
15							15	17	17	6	3	1*	R
16							16	17	17	6*	4	2	R
17							17	18	18	1	2	1	A
18						12		11	16	4	2	2*	R
19						13		12	17	3*	4	1	R
20						14		12	18*	5	6	2*	R,=VT
21					6			7	13	1	2	2*	R
22					7			8	14	1	5	1	A
23						8		11	14	2	3*	1	R
24						9		11	14	6	1*	1	R
25						10		11	15	5	4	2	A
26							11	15	15	2	5	2*	R
27							12	15	15	4	2	2*	R
28							13	16	16	3*	4	1	R
29							14	16	16	5	6	2*	R

30							15	17	17	6	3	1*	R
31							16	17	17	6*	4	2	R
32							17	18	18*	1*	2	1	R,=VT
33						11		12	16	2	5	2*	R
34						12		12	17	4	2	2*	R
35						13		13	18*	3	4	1	R,=VT
36					8			8	14	2	3*	1	R
37					9			8	15	6	1*	1	R
38					10			8	16	5	4	2	A
39						11		12	16	2	5	2*	R
40						12		12	17	4	2	2*	R
41						13		13	18*	3	4	1	R,=VT
42					11			9	18*	2	5	2*	R,=VT
43				5				5	13	5	6	1	A
44					6			7	13	1	2	1*	R
45					7			8	14	1	5	2*	R
46					8			8	14	2	3*	1	R
47					9			8	15	6*	1	1	R
48					10			8	16	5*	4	2	R
49					11			9	18*	2	5	2	R,=VT
50				6				5	14	1	2	2*	R
51				7				6	15	1	5	1	A
52					8			9	15	2	3*	1	R
53					9			9	16	6*	1	1	R
54					10			9	17	5	4	2	A
55						11		13	17	2	5	2*	R
56						12		13	18*	4	2	2*	R,=VT
57					11			10	19*	2	5	2*	R,>VT
58				8				6	16	2	3*	1	R
59				9				6	17	6*	1	1	R
60				10				6	19*	5	4	2	R,>VT
61			4					4	14	3	1	1	A
62				5				6	14	5	6	1	A
63					6			8	14	1	2	2*	R
64					7			9	15	1	5	1*	R
65					8			9	15	2	3*	1	R
66					9			9	16	6	1*	1	R
67					10			9	17	5*	4	2	R
68					11			10	19*	2	5	2	R,>VT
69				6				6	15	1	2	2*	R
70				7				7	16	1	5	1	A
71					8			10	16	2	3	1*	R
72					9			10	17	6	1*	1	R
73					10			10	18*	5	4	2	R
74				8				7	17	2	3	1*	R
75				9				7	18*	6	1	1	R
76			5					4	15	5	6	1	A
77				6				6	15	1	2	2*	R
78				7				7	16	1	5	1	R
79				8				7	17	2	3	1*	R
80				9				7	18*	6	1	1	R,=VT
81			6					4	16	1	2	2*	R
82			7					5	18*	1	5	1	R
83		3						2	11	6	3	2	A
84			4					4	11	3	1	1	A
85				5				6	11	5	6	1	A
86					6			8	11	1	2*	2	R
87					7			9	12	1	5	1	A
88						8		12	12	2	3	1*	R
89						9		12	12	6	1*	1	R

90						10		12	16	5	4	2	A
91							11	16	16	2	5	2	A
92						11		13	17*	2	5	2	R,>VT
93					8			9	12	2	3*	1	R
94					9			9	12	6	1*	1	R
95					10			9	17*	5	4	2	R,>VT
96				6				6	12	1	2	2*	R
97				7				7	13	1	5	1	A
98					8			10	13	2	3*	1	R
99					9			10	13	6	1*	1	R
100					10			10	14	5	4	2	A
101						11		14	18	2	5	2	R,>VT
102					11			11	20	2	5	2	R,>VT
103				8				7	13	2	3*	1	R
104				9				7	14	6	1*	1	R
105				10				7	15	5	4	2	A
106					11			11	20*	2	5	2	R,>VT
107				11				8	17*	2	5	2	R,>VT
108			5					4	12	5	6	1	A
109				6				6	15	1	2	2	A
110					7			9	15	1*	5	1	R
111					8			9	15	2	3*	1	R
112					9			9	16*	6	1*	1	R,=VT
113				7				7	13	1	5	1*	R
114				8				7	13	2	3	1*	R
115				9				7	14	6*	1	1	R
116				10				7	15	5	4	2	A
117					11			11	20*	2	5	2	R,>VT
118				11				8	17*	2	5	2	R,>VT
119			6					4	16*	1	2	2	R,=VT
120		4						3	13	3	1	1	A
121			5					5	13	5	6	1	A
122				6				7	16*	1	2	2	R,=VT
123			6					5	17*	1	2	2	R,>VT
124		5						3	14	5	6	1	A
125			6					5	17*	1	2	2	R,>VT
126		6						3	19*	1	2	2	R,>VT
127	2							1	11	2	4	2*	R
128	3							2	14	6	3	2	A
129		4						4	14	3	1	1	A
130			5					6	14	5	6	1	A
131				6				8	14	1	2	2	A
132					7			11	14	1*	5	1	R
133					8			11	14	2	3	1*	R
134					9			11	14	6*	1	1	R
135					10			11	15	5*	4	2	R
136					11			12	16*	2	5	2*	R,=VT
137				7				9	15	1	5	1*	R
138				8				9	19*	2	3	1	R,>VT
139			6					6	18*	1	2	2	R,>VT
140		5						4	16*	5	6	1	R,=VT
141	4							2	15	3	1	1	A
142		5						4	15	5	6	1	A
143			6					6	15	1	2	2	A
144				7				9	15	1*	5	1	R
145				8				9	15	2	3*	1	R
146				9				9	16*	6	1	1	R,=VT
147			7					7	16*	1	5	1*	R,=VT
148		6						4	16*	1	2	2	R,=VT
149	5							2	16*	5	6	1	R,=VT

From the above **Table – 7**, gives optimal solution of the taken numerical example and the word is L_7 = (3, 4,5, 7, 9,10,11) is a optimal feasible word. The set of ordered triplets which satisfy the optimum solution (4,2,1),(6,3,2),(3,1,1),(5,6,1),(2,5,2),(1,5,1),(5,4,2). At the end of search table optimal solution **VT** is **16**. It is in 91^{st} row of the search table. For this optimal feasible word the array L, IR, IC, SW and IK are given in the following Table – 8.

Table8.

	1	2	3	4	5	6	7
L	**1**	**3**	**4**	**5**	**7**	**10**	**11**
IR	**1**	**1**	**1**	**1**	**1,1**	**1**	
IC	**1**	**1**	**1**	**1**	**1,1**	**1**	
SW	**5**	**5**	**1**	**2**	**6,4**	**3**	
IK	**1**	**2**	**1**	**1**	**1,2**	**2**	

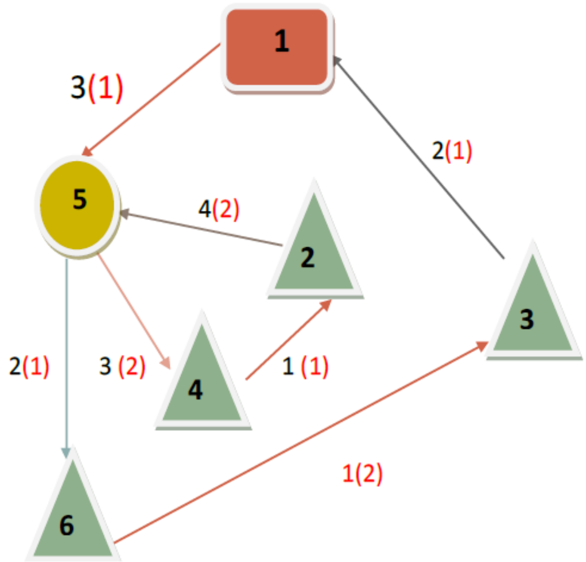

Fig2.

In the above figure-2, rectangle represents head quarter city, circle indicates revisiting city and triangles represents remaining cities in salesman tour. The value in rectangle/triangle/circle indicates name of the city. Also value at each arc in parenthesis represents the facility used and before parenthesis represents distance between respective two cities.

At the end of the search, the current value of the VT is 16 and it is the value of optimal feasible word L_7 = (1,3,4,5,7,10,11), it is given in the91^{st} row of the search table. So, value of optimal solution of the model "Generalized TSP model with clusters" by Lexi search algorithm using pattern recognition technique is 16.

Z=D(4,2,1)+D(6,3,2)+D(3,1,1)+D(5,6,1)+D(2,5,2)+D(1,5,1)+D(5,4,2)

=1+1+2+2+4+3+3=16

Consider the set of ordered triples{(4,2,1),(6,3,2),(3,1,1),(5,6,1),(2,5,2),(1,5,1),(5,4,2)} represented the pattern given in the tables-10 & 11 which is an optimal solution. According to the pattern represented in figure-2 satisfies all the constraints in the mathematical formulation.

Table-9

D(i,j,1)

$$\begin{bmatrix} 0&0&0&0&1&0 \\ 0&0&0&0&0&0 \\ 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&0&0&0&1 \\ 0&0&0&0&0&0 \end{bmatrix}$$

Table-10

D(i,j,2)

$$\begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&1&0&0&0 \end{bmatrix}$$

The feasible word is L_7=(1,3,4,5,7,10,11) is a feasible word. For this optimal word the array IR, IC, L, SW are given in the above table-7

12. Conclusion

In this paper, we presented an exact algorithm called Lexi-search algorithm based on pattern recognition technique to solve the TSP Problem. Lexi-search algorithms are proved to be more efficient in many combinatorial problems. First the model is formulated into a zero-one programming problem. A Lexi-Search Algorithm based on Pattern Recognition Technique is developed for getting an optimal solution. We strongly consider that this algorithm can perform larger size problems.

References

[1] Kubo,M. &Kasugai,H., The precedence constrained TSP, Journal of Operations research of Japan, 34,152.

[2] Das,S., The most economical route for a TSP, paper presented by in ORSI, 1978.

[3] Pandit,S.N.N., An intelligent search approach to TSP Symposium in OR, IIT, Kharagpur.

[4] Ramesh,T., Travelling purchaser problem,Operations research,18(2),781.

[5] Raviganesh, Murthy,G.S.,Das, S., Solution of TSP in a protean network, IJOMAS,14(1),99.

[6] Srivastava, S., Santhosh kumar, S., Garg,R.C. & Sen,P.,Generelized TSP through n sets of nodes, Operaations research,7(2), 97-101.

[7] Little,J.D.C., Murthy, K.G. & Sweeny, D.W., An algorithm for the TSP, Operations research,11,972-982.

[8] Sundara Murthy,M(1979):Combinatorial Programming-A Pattern Recognition approach., REC, Warangal, India

[9] Hardgrave,W.W & G.L.Nambhauser(1962):On the Relation between the Travelling Salesman and the Longest Path Path Problems, Operations Research,10(647).

[10] Flood,M.M.(1956):The TSP Operations Research,41(61-75)