

Three Dimensional Constrained P-Tsp

Sreenivasulu Y¹, Vijayalakshmi R², Somasekhar Srinivas .V K³

Sundaramurthy M⁴

¹Research Scholar, Dept., of Mathematics, S.V.University, Tirupati

²Research Scholar, Dept., of Mathematics, S.V.University, Tirupati

³Assistant professor, Dept., of Mathematics, Annamacharya Institute of Technology and Sciences, Tirupati

⁴Rtd., Professor, Dept., of Mathematics , S.V.University, Tirupati

Abstract: Many Combinatorial programming problems are NP-hard (Non Linear Polynomial), and we consider one of them called THREE DIMENSIONAL CONSTRAINED P-TSP. Let there be n cities and the distance/cost matrix $D(i, j, k)$ is given from i^{th} city to j^{th} city in k^{th} season. Each of P salesmen starts in first season from Headquarter city '1' and visits few cities in that season then he enter into next season and visits some more cities in that season and reaches to Headquarter city on or before the last season. The aim of the problem is to find p tours to the p salesmen which covers $n-1$ cities and the total distance /cost of the tours is minimum subject to the considerations. For this problem we developed a Lexi Search Algorithm using Pattern Recognition Technique.

Keywords: Lexi Search Algorithm, Pattern Recognition Technique, Tour, Pattern, Season.

I. Introduction

The 'Travelling Salesman problem' [TSP] formulated in literature as:

“Suppose a salesman wants to visit a certain number of cities allotted to him. He knows the distance/cost and time of journey between every pair of city 'i' and city 'j' denoted c_{ij} . The problem is to select a route that starts from a given home city to passes through each and every city once and only once in the shortest distance.” The "Travelling Salesman Problem"[TSP] has become a wide research concept in mathematics and computers by experts, scientists and scholars, with aim of minimizing the total travel distance of the salesman in Travelling Salesman Problem [TSP]. Generalized Travelling Salesman Problem was mentioned in [13,22,24]. Exact algorithms found in [14, 11]. Several standard heuristic approaches are applied to solve the Travelling Salesman Problem.

The standard travelling salesman problem can be called “Two- Dimensional Travelling Salesman Problem”. The Time Dependent Travelling Salesman problem [TDTSP] is a Generalization from the Travelling salesman problem. The TDTSP mentioned as a scheduling problem. The TDTSP earlier was attempted by [16]. Several authors studied related problem in machine scheduling by [20], [3,27] a new formulation of TDTSP which uses n^3 variables and only n constraints, simulated annealing heuristics proposed by [26]. Several properties and algorithms for Time –Dependent Shortest path problem [TDSPP] have been reviewed by [1,7,8,19] among others. [12] addressed a (2+ ϵ) approximation algorithm for Time Dependent orienteering problem which runs in polynomial time if the ratio between the maximum and minimum travel times between any two cities is constant. [2] addressed as TDTSP considered a variant with time windows. [15] Improved a dynamic programming algorithm for TDTSP explains a user specified number of partial tours is retained at each stage. Moving target TSP describes a pursuer must intercept in minimum time a set of targets that move with constant velocities. A genetic algorithm also developed for above problem. [17] explained the robust TSP with data in which time of travel are mentioned as ranges of possible values regarding lexicographic search in Time dependent travelling salesman problem [TDTSP] was also attempted by [5&25]. [18] addressed a Continuous Time Dependent Travelling Salesman Problem describes, the cost is minimum, if the cost of the salesman visits from city i to city j at a point time k [1, 2, 3... n] which is continuous he should not visit more than one pair of cities is minimum. [4] addressed two problem on [TDTSP] (A) Generalized Time Dependent Travelling Salesman Problem with Cyclic constraint [GTSP] describes cost is minimum, if the salesman travel from city i to city j at time $[k]$ as a facility which influences the cost, when set of n cities divided into different grades and he has to travel p cities [$p < n$]. (B) Generalized Time Dependent Travelling Salesman Problem (cluster constraint) [GTDTSP] the cost is minimum, if the sales man visits city i to city j at time (availing facility) which influence the cost when he travelled 'm' cities in a cluster [$m < n$] out of total 'n' cities. In this paper we study a variant TSP model.

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. $A = (1,2,3,\dots, n)$ and the different possible sequences of length k of these symbols. Thus $(\alpha_1, \alpha_2, \dots, \alpha_k)$ 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 a 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]. In 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 recognizing 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 two 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, Vidhyulata – 1992, Ramana and Umashankar -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.

In this section we discuss a model which is a variant model of **minimum spanning models** called **“THREE DIMENSIONAL CONSTRAINED P-TSP. Generally distance from i^{th} city to j^{th} city is mentioned in matrix $D(i, j)$. But here we introduce an the season which influences the distance/cost and that factor is represented as a season, which is denoted by k . Therefore here D is the given distance/ cost matrix, $D(i, j, k)$ be the distance/cost from i^{th} city to j^{th} city with season k . Let there are n cities, the given distance matrix $D(i, j, k)$ represents distance from i^{th} city to j^{th} city in k^{th} season . It is a three dimensional problem. Let there be few say P salesmen. Each salesman starts in first season from Headquarter city and visits few cities in that season then he enter into next season and visit some more cities in that season and reaches to Headquarter city on or before last season. The aim of the problem is to find P tours with minimum total distance which covers $n-1$ cities by the salesmen subject to the above considerations.**

Let a set $N = \{1,2,3, \dots, n\}$ where n is number of cities and the set of k seasons $K = \{1,2,3, \dots, k\}$. let $D(i,j,k)$ be the distance\cost from i^{th} city to j^{th} city in k^{th} season where $i, j \in N$ and $k \in K$. Let $\{1\}$ be the head quarter city and P be the number of sales men. . Each salesman starts in first season from Headquarter city and visits few cities in that season then he enter into next season and visits some more cities in that season and reached to Headquarter city on or before last season. The aim of the problem is to find P tours with total minimum distance which covers all $n-1$ cities by the salesmen subject to the above

considerations . For this we developed an algorithm called as Lexi search algorithm using pattern recognition technique and it is illustrated with a suitable numerical example with two travelling sales man and three seasons.

Mathematical Formulation:

$$\text{Minimize } Z(X) = \sum_i \sum_j \sum_k D(i,j,k) X(i,j,k) \text{ -----(1)}$$

where $i, j \in N$ & $k \in K$

$$\left. \begin{aligned} \sum_j \sum_k X(1,j,k) &= P, \quad \text{where } j \in N, k \in K \\ \sum_i \sum_j X(i,1,k) &= P, \quad \text{Where } i \in N, k \in K \end{aligned} \right\} \text{----- (2)}$$

$$\left. \begin{aligned} n_i &= |N_i| \text{ represents number of cities in } i^{\text{th}} \text{ tour of the } i^{\text{th}} \text{ salesman} \\ \bigcap_{i=1}^p N_i &= \{1\} \text{ and } \bigcup_i N_i = N \end{aligned} \right\} \text{-----(3)}$$

$$\left. \begin{aligned} \sum_{i=1}^N n_i &= n+p-1 \\ \sum_i \sum_j \sum_k X(i,j,k) &= n+p-1 \end{aligned} \right\} \text{-----(4)}$$

$$\text{There is a tour in } N_i \text{ cities by } i^{\text{th}} \text{ sales man who starts in } \{1\} \text{----- (5)}$$

$$\left. \begin{aligned} X(i_1, j_1, k_1) &= X(j_1, j_2, k_2) = 1, \quad i=1,2, \dots, N \\ k_1 \leq k_2, \quad j_1 &\neq 1, \quad \forall j_1 \in N_i \text{ and } j_2 \in N_i, \quad k_1, k_2 \in K \end{aligned} \right\} \text{----(6)}$$

$$X(i,j,k) = 1 \text{ or } 0 \text{ -----(7)}$$

Equation (1) represents that the objective function of the problem, i.e. to find total minimum distance for the tours by salesmen of all the cities from head quarter city.

Equation (2) indicates the P-travelling salesmen starts from Head Quarter city & finally reaches to Headquarter city .

Equation (3) shows that the number of cities travelled by each salesman in their tour . i.e N_i for i^{th} salesman .

Equation (4) gives the total number of connections in the tour/Network.

Equation (5) shows that there is a tour in the N_i cities starting from city 1 by the i^{th} salesmen.

Equation (6) shows that a sales man reaches a city $j_1 \in N_i$ from a city in N_i in the season k_1 and further goes to another city $j_2 \in N_i$ in K_2 season then $k_1 \leq k_2$. i.e in the tour of sealesman the season he visits a city should be always less than or equal to the season when leaves the city.

Equation (7) indicates, if a city i is connected to city j in k^{th} season then $X(i,j,k)=1$, otherwise it will be equal to 0.

II. Numerical Illustration

The concepts and algorithm developed will be illustrated by a numerical example for which total number of cities $N=\{1,2,3,4,5,6,7,8\}$. There are three seasons and two salesmen. The distance/cost matrices $D(i,j,k)$ are as follows

Table-1
D(i,1)

SN	1	2	3	4	5	6	7	8
1	---	23	1	21	8	7	-	2
2	25	---	43	19	27	1	28	31
3	41	2	---	44	24	-	32	6
4	14	47	22	---	42	26	5	13
5	48	3	39	40	---	20	-	30
6	46	15	45	9	29	---	-	12
7	33	-	10	-	11	17	---	38
8	35	16	34	3	18	36	37	---

Table-2
D(i,2)

SN	1	2	3	4	5	6	7	8
1	---	23	18	2	33	16	30	26
2	35	---	4	34	31	47	3	29
3	36	48	---	5	7	27	25	28
4	20	32	6	---	3	22	46	24
5	1	7	3	38	---	15	37	44
6	-	12	8	-	14	---	45	43
7	-	39	11	21	9	41	---	-
8	40	-	19	10	-	49	42	---

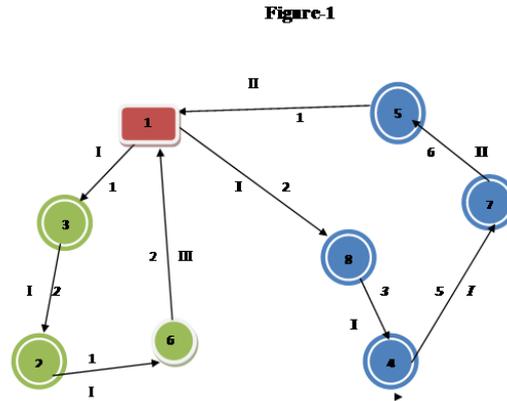
Table-3
D(i,3)

SN	1	2	3	4	5	6	7	8
1	---	21	1	46	19	36	11	38
2	20	---	47	3	15	5	49	26
3	18	2	---	35	31	12	22	27
4	40	16	37	---	44	14	39	42
5	45	41	7	43	---	28	24	25
6	2	17	32	30	29	---	48	-
7	-	-	9	34	51	4	---	-
8	-	10	33	13	23	50	-	---

In the above tables (tables-1,2&3) the value ‘∞’ & ‘-’ indicates irrelevant and the non-connectivity of the cities .Here the entire D(i,j,k)’s are taken as positive integers which is not necessary always . In table- 1, suppose D(7,8,1)=38 means the distance /cost of the connecting the city 7 to 8 in season I is 38.

Feasible Solution:

Consider a set order triples (1,3,1),(3,2,1),(2,6,1),(6,1,3),(1,8,1),(8,4,1),(4,7,1), (7,5,2),(5,1,2) represents a feasible solution. In the following **figure-1**, square represents Headquarter city {1},circle represents city and value in circle indicates number of that particular city. A value below and above the connectivity of the city represents distance between the two cities and respective season between two particular cities.



From the above figure-1, In a tour ,the first sales man starts from city 1 and visits city 3 in season I , from city 3 he visits city 2 in season I , from city 2 he visits city 6 in season I, from city 6 he returns to city 1 in season III. The second salesman starts from city 1 and visits city 8 in season I, from city 8 he visits city 4 in season I , from city 4 he visits city 7 in season I , from city 7 he visit city 5 in season II , from city 5 he returns to city1 in season II. In this way all the cities i.e 8 cities are covered by the salesmen . Hence value of the the solution is as follows

$$\begin{aligned}
 Z &= D(1,3,1)+D(3,2,1) + D(2,6,1) + D(6,1,3) + D(1,8,1)+ D(8,4,1)+ D(4,7,1)+ (7,5,2)+ D(5,1,2) \\
 &= 1+2+1+2+2+3+5+6+1 \\
 &= 23
 \end{aligned}$$

Solution Procedure:

In the above **figure-1**, for the feasible solution we observed that 9 **ordered triples** are taken along with their values from the cost/distance matrices for the numerical example in **table-3**. The 9 ordered triples are selected such that they represents a feasible solution according **figure-1**. So the problem is that we have to select 9 ordered triples from the cost/distance matrices [8x8x3] along with values such that the total cost/distance is minimum and represents a feasible solution. For this selection of 9 ordered triples we arranged **the** ordered triples in the increasing order and call this formation as **alphabet table** and we developed an algorithm for the selection along with the checking for the feasibility.

III. Concepts and definitions

Definition of Pattern:

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

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

The value V(X) gives the total cost/ distance of the solution represented by X. Thus X is the feasible pattern gives the total cost/ 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 triples (i, j, k) for which X (i, j, k) =1, with understanding that the other X (i, j, k) are zeros.

Table-4: Feasible pattern

X(i, j, 1)							
0	0	1	0	0	0	0	1
0	0	0	0	0	1	0	0
0	1	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	1	0	0	0	0

X(i, j, 2)							
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
1	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

X(i, j, 3)							
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
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

In the above solution X(3, 2, 1) = 1 represents that a sales man travel from city3 to city2 in season I. In similar way X(7, 5, 2) = 1 represents a sales man travel from city 7 to city 5 in season II. Similarly all the cities are traveled by the sales men from head quarter {1}. So, the above solution gives a feasible solution and it shown in figure-1.

Alphabet Table:

There are n x n x r ordered triples in the three-dimensional array D. For convenience these are arranged in ascending order of their corresponding cost/distance and are indexed from 1, 2, 3,.... (Sundara Murthy-1979). Let

SN=(1, 2, 3,....) be the set of indices. Let D be the corresponding array of cost/distance. For convenience same notation 'D' is used. If a, b ∈ SN and a < b then D(a) ≤ D(b). Also let the arrays R, C & K be the array of row, column and season indices. CD be an 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 a ∈ SN then (R(a), C(a), K(a)) is the ordered triple and D(a) = D(R(a), C(a), K(a)) is the value of the ordered triple and CD(a) = ∑_{i=1}^a D(i).

Table-5
ALPHABET TABLE

S N0	D	CD	R	C	K
1	1	1	1	3	1
2	1	2	2	6	1
3	1	3	5	1	2
4	1	4	1	3	3
5	2	6	1	8	1
6	2	8	3	2	1
7	2	10	1	4	2
8	2	12	3	2	3
9	2	14	6	1	3
10	3	17	5	2	1
11	3	20	8	4	1
12	3	23	2	7	2
13	3	26	4	5	2
14	3	29	2	4	3
15	4	33	2	3	2
16	4	37	7	6	3
17	5	42	4	7	1
18	5	47	4	7	1
19	5	52	2	6	3
20	6	58	3	8	1
21	6	64	7	5	2
22	7	71	1	6	1
23	7	78	5	2	2
24	7	85	5	3	3
25	8	93	1	5	1
26	8	101	6	3	2
27	9	110	6	4	1
28	9	119	4	3	2
29	9	128	7	3	3

30	10	138	7	3	1
31	10	148	8	4	2
32	10	158	8	2	3
33	11	169	7	5	1
34	11	180	7	3	2
35	11	191	1	7	3
36	12	203	6	8	1
37	12	215	6	2	2
38	12	227	3	6	3
39	13	240	4	8	1
40	13	253	5	3	2
41	13	266	8	4	3
42	14	280	4	1	1
43	14	294	6	5	2
44	14	308	4	6	3
45	15	323	6	2	1
46	15	338	5	6	2
47	15	353	2	5	3
48	16	368	8	2	1
49	16	384	1	6	2
50	16	400	4	2	3
51	17	417	7	6	1
52	17	434	3	5	2
53	17	451	6	2	3
54	18	469	8	5	1
55	18	487	1	3	2
56	18	505	3	1	3
57	19	524	2	4	1
58	19	543	8	3	2
59	19	562	1	5	3
60	20	582	5	6	1
61	20	602	4	1	2
62	20	622	2	1	3
63	21	643	1	4	1
64	21	664	7	4	2
65	21	685	1	2	3
66	22	707	4	3	1
67	22	729	4	6	2
68	22	751	3	7	3
69	23	774	1	2	1
70	23	797	1	2	2
71	23	820	8	5	3
72	24	844	3	5	1
73	24	868	4	8	2
74	24	892	5	7	3
75	25	917	2	1	1
76	25	942	3	7	2
77	25	967	5	8	3
78	26	993	4	6	1
79	26	1019	1	8	2
80	26	1045	2	8	3
81	27	1072	2	5	1
82	27	1099	3	6	2
83	27	1126	3	8	3
84	28	1154	2	7	1
85	28	1182	3	8	2
86	28	1210	5	6	3
87	29	1239	6	5	1
88	29	1268	2	8	2
89	29	1297	6	5	3
90	30	1327	5	8	1
91	30	1357	1	7	2
92	30	1387	6	4	3
93	31	1418	2	8	1
94	31	1449	2	5	2
95	31	1480	3	5	3
96	32	1512	3	7	1
97	32	1544	4	2	2
98	32	1576	6	3	3
99	33	1609	7	1	1

100	33	1642	1	5	2
101	33	1675	8	3	3
102	34	1709	8	3	1
103	34	1743	2	4	2
104	34	1777	7	4	3
105	35	1812	8	1	1
106	35	1847	2	1	2
107	35	1882	3	4	3
108	36	1918	8	6	1
109	36	1954	3	1	2
110	36	19990	1	6	3
111	37	2027	8	7	1
112	37	2064	5	7	2
113	37	2101	4	3	3
114	38	2139	7	8	1
115	38	2177	5	4	2
116	38	2215	1	8	3
117	39	2254	5	3	1
118	39	2293	7	2	2
119	39	2332	4	7	3
120	40	2372	5	4	1
121	40	2412	8	1	2
122	40	2452	4	1	3
123	41	2493	3	1	1
124	41	2534	7	6	2
125	41	2575	5	2	3
126	42	2617	4	5	1
127	42	2659	8	7	2
128	42	2701	4	8	3
129	43	2744	2	3	1
130	43	2787	6	8	2
131	43	2830	5	4	3
132	44	2874	3	4	1
133	44	2918	5	8	2
134	44	2962	4	5	3
135	45	3007	6	3	1
136	45	3052	6	7	2
137	45	3097	5	1	3
138	46	3143	6	1	1
139	46	3189	4	7	2
140	46	3235	1	4	3
141	47	3282	4	2	1
142	47	3329	2	6	2
143	47	3376	2	3	3
144	48	3424	5	1	1
145	48	3472	3	2	2
146	48	3520	6	7	3
147	49	3569	8	6	2
148	49	3618	2	7	3
149	50	3668	8	6	3
150	51	3719	7	5	3

Let us consider 18ESN. It represents the ordered triple $(R(18), C(18), K(18)) = (3, 4, 2)$. Then $D(18) = D(3, 4, 2) = 5$ and $CD(18) = 47$.

Definition a Word:

Let $SN = (1, 2, \dots)$ be a set of indices, let D be an array of cost/distance, CD is an array of cumulative sum of elements in D . Let arrays R , C and K be the row, column and season indices of the ordered triples. Let K be the array of season. Let $L_k = \{a_1, a_2, \dots, a_k\}$, $a_i \in SN$ 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 $a_i \leq a_{i+1}$, $i = 1, 2, \dots, k-1$. The set SN is defined as the “Alphabet-Table” with alphabet order as $(1, 2, \dots)$ 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 $i = 1, 2, \dots, k-1$ and if this condition is not met it is called a “insensible word”. A word L_k has at least one feasible word or equivalently the partial pattern represented by L_k has at least one feasible word or equivalently the partial pattern represented by 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 ' $n+P-1$ '. Since the words of length greater than $n+P-1$ are necessarily infeasible, as any feasible pattern can have only $n+p-1$ unit entries in it. If $k \leq n+P-1$, L_k is called a partial word and if $k=n+P-1$, 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 letters. A partial word 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.

Value of the Word:

The value of the partial word L_k , $V(L_k)$ is defined recursively as $V(L_k)=V(L_{k-1})+D(a_i)$ with $V(L_0)=0$, where $D(a_i)$ is the distance array $V(L_k)$ and $V(X)$ the values of the pattern X will be the same, since X is the (partial) pattern represented by L_k , (Sundara Murthy – 1979).

Consider the partial word $L_4 = (1,2,3,5)$
 Then $V(L_4) = 1+1+1+2=5$

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 $L_k = (a_1, a_2, \dots, a_k)$ can be defined as follows.

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

Consider the partial word $L_4 = (1, 2, 3, 5)$ and $V(L_4) = 1+1+1+2=5$ Then $LB(L_4) = V(L_4) + CD(a_4 + n + 2 - 1 - 4) - CD(a_4)$

$$LB(4) = 5 + CD(5+8+1-4) - CD(5)$$

$$= 5 + CD(10) - CD(5)$$

$$LB(4) = 5 + 17 - 06 = 16$$

Feasibility criterion of partial word:

An algorithm was developed, in order to check the feasibility of a partial word $L_{k+1} = \{a_1, a_2, \dots, a_k, a_{k+1}\}$ 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 $IR(i)=1, i \in \mathbb{N}$ indicates that from the i^{th} city a salesman visits some city in his tour, otherwise $IR(i)=0$
- ❖ **IC** be an array where $IC(i) = 1, i \in \mathbb{N}$ indicates that a salesman visits from a city in his tour to i^{th} city, otherwise $IC(i)=0$
- ❖ **SW** be an array where $SW(i) = j$ indicates that a salesman visits from i^{th} city to city j, otherwise $SW(i) = 0$
- ❖ **L** be an array where $L[i] = a_i, i \in \mathbb{N}$ is the letter in the i^{th} position of a word
 Then the values of the arrays IR, IC, SW and L are as follows
- ❖ $IR(R(a_i)) = 1, i = 1, 2, \dots, k$ and $IR(j) = 0$ for other elements of j
- ❖ $IC(C(a_i)) = 1, i = 1, 2, \dots, k$ and $IC(j) = 0$ for other elements of j
- ❖ $SW(R(a_i)) = C(a_i), i = 1, 2, \dots, k$ and $SW(j) = 0$ for other elements of j
- ❖ $L(i) = a_i, i = 1, 2, \dots, k$, and $L(j) = 0$, for other elements of j
- ❖ For example consider a feasible partial word $L_6 = (1, 2, 3, 5, 6, 9)$ which is feasible. The array IR, IC, L, SW takes the values represented in **Table -6** given below.

Table-6

	1	2	3	4	5	6	7	8
L	1	2	3	5	6	9		
IR	1+1	1	1		1	1		
IC	1+1	1	1			1		1
SW	3,8	6	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 $TR=R(a_i), TC=C(a_i) \ \& \ TK=K(a_i)$.

IV. Feasible Algorithm

```

102  IX=0      GO TO 102
104  IS IR(RA) =HE  YES(IR(RA) =IR(RA)+1)
GO TO      106      NO_108
106  IS IR(RA) ≤Z  YES 108 , NO_END_
108  IS IC(CA) =HC  YES_IC(CA) =IC(CA)+1_ ,
      GO TO 110      NO__112_
110  IS IC(CA) ≤Z  YES_112_ ,      NO_END
112  IS IR(RA) =0  YES_114_ ,      NO_END_
114  IS IC(CA) =0  YES_116_ ,      NO_END_
116  W =CA        GO TO 118
118  IS SW(W) =0   YES_124_ ,      NO_120
120  IS SW(W) =RA  YES_END_ ,      NO_122_
122  W =SW(W)     GO TO 118
124  IS SW[CA(KA)] =0  YES_126_ ,      NO_128
126  IS KA >SW[CA(KA)] =0  YES_130_ ,      NO END
128  IS KA ≤ SW[CA(KA)]  YES 130      NO
130  W =RA        GO TO 132
132  IS SWI(W) =0   YES 138 ,      NO 134
134  IS SWI(RA) =CA  YES_END_ ,      NO__136
136  W =SWI(W)     GO TO__132
138  IS SWI[RA(KA)] =0  YES__140_ ,      NO__140
140  IS KA ≥ SWI [RA(KA)]  YES __142_ ,      NO_END
142  W =HC        GO TO 144
144  IS SW(HC) =0  YES__152_ ,      NO__146
146  IS SW(HC) =HC  YES __150__` ,      NO__148
148  W =SW(HC)    GO TO 144
150  LP1 =1       GO TO 152
152  IS I=N       YES__154_ ,      NO__162
154  IS SWI(HC) =0  YES__S__ ,      NO_ 156
156  IS SWI(HC) =HC  YES__160_ ,      NO_158
158  W =SWI(HC)   GO TO 154
160  LP2 =1       GO TO 162
162  IX =1
164  STOP/END

```

$L_k = L_{k-1} * a_k$ 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 processed to consider a partial word of $k+1$. Which represents a sub-block of the block of Words represented by L_k .

$LB(L_k) \geq VT$. 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 .

Lexi-Search Algorithm

```

2  I=1
   J=J+1
   IS J>MAX      YES___ ,      NO_3__
3  RA=R(J)
   CA=C(J)
   KA=K          GO TO 4
4  FEASIBLE CHEACKING
   IS IX=1      YES__6_ ,      NO__2
6  V(I)=V(I-1) +D(J)
   LB(I)=V(I) +CD(j+n+1-a_k) -CD(J)
   IS LB(I) ≥ VT  YES__11_ ,      NO__7
7  IS LP1=1 & LP2=1      YES_9_ ,      NO__8
8  IR(RA) = IR(RA)+1

```

```

IC(CA) = IC(CA)+1
SW(RA) = CA
SWI(CA) = RA
  IK(KA) = KA          GO TO __8A
8A  I=I+1             GO TO 2
9   V(I) =VT, RECARD V   GO TO __10
10  I=I+1             GO TO __11
11  IS I=1            YES __13_, NO__12
12  J=L(I)
    IR(RA) =IR(RA)-1
    IC(CA) = IC(CA) -1
    SW(RA) =0
    SWI(CA) =0
    IK(KA) =0          GO TO ____8A
13  STOP .
    
```

Search Table:

The working details of getting an optimal word using the above algorithm for the illustrative numerical example is given in the following table-6. The columns named (1), (2),(3).....gives the letters in the first, second, third and so on places respectively. The columns V, LB, R, C and K letters indicates Value of that word ,lower bound of the word , row, column and season indices of accepted for connectivity. The last column denotes the remarks regarding the acceptability of the partial word. In the following table ‘A’ indicates ACCEPT and ‘R’ indicates REJECT.

Table---7 Search Table

	1	2	3	4	5	6	7	8	9	V	LB	R	C	K	Remarks
1	1									1	14	1	3	1	A
2.		2								2	14	2	6	1	A
3.			3							3	14	5	1	2	A
4.				4						4		1	3	3	R
5.					5					5	16	1	8	1	A
6.						6				7	16	3	2	1	A
7.							7			9		1	4	2	R
8.								8		9		3	2	3	R
9.									9	9	18	6	1	3	A
10.								10		12		5	2	1	R
11.									11	12	18	8	4	1	A
12.								12		15		2	7	2	R
13.									13	15		4	5	2	R
14.								14		15		2	4	3	R
15.									15	16		2	3	2	R
16.									16	16		7	6	3	R
17.								17		17	22	4	7	1	A
18.									18	22		3	4	2	R
19.									19	22		2	6	3	R
20.									20	23		3	8	1	R
21.									21	23	23	7	5	2	A=VT
22.								18		17		3	4	2	R
23.								19		17		2	6	3	R
24.								20		18		3	8	1	R
25.									21	18	25	7	5	2	R _c >VT
26.							12			12		2	7	2	R
27.								13		12	19	4	5	2	A
28.								14		15		2	4	3	R
29.								15		16		2	3	2	R
30.								16		16		7	6	3	R
31.								17		17		4	7	1	R
32.								18		17	22	3	4	2	R
33.								19		17	22	2	6	3	R
34.								20		18	24	3	8	1	R _c >VT
35.							14			12		2	4	3	R
36.							15			13		2	3	2	R
37.							16			13		7	6	3	R
38.								17		14	24	4	7	1	R _c >VT
39.							10			10		5	2	1	R
40.								11		10	19	8	4	1	A
41.							12			13		2	7	2	R

148.					11					12	21	8	4	1	A
149.										12		2	7	2	R
150.										13		4	5	2	R
151.										14		15	23	2	R _t =VT
152.										12		2	7	2	R
153.										13		12	23	4	R _t =VT
154.						11					9	22	8	4	A
155.										12		2	7	2	R
156.										13		12	23	4	R _t =VT
157.										12		9	23	2	R _t =VT
158.						10					7	23	5	2	R _t =VT
159.						6					4	19	3	2	A
160.						7					6		1	4	R
161.						8					6		3	2	R
162.						9					6	21	6	1	A
163.										10		9	5	2	R
164.										11		9	22	8	A
165.										12		12	2	7	R
166.										13		12	23	4	R _t =VT
167.										12		9	23	2	R _t =VT
168.										10		7	23	5	R _t =VT
169.										7		4		1	R
170.										8		4		3	R
171.										9		4	23	6	R _t =VT
172.										3		2	16	5	A
173.										4		3		1	R
174.										5		4	18	1	A
175.										6		6	18	3	A
176.										7		8		1	R
177.										8		8		3	R
178.										9		8	20	6	A
179.										10		11		5	R
180.										11		11	20	8	A
181.										12		14	20	2	A
182.										13		17	20	4	A
183.												14	20	2	R
184.												15	21	2	R
185.												16	21	7	A _t =VT
186.												17	21	2	R _t =VT
187.												14	21	4	R _t =VT
188.												11	21	2	R _t =VT
189.												9	21	5	R _t =VT
190.												6		1	R
191.												6	20	3	A
192.												8	20	6	A
193.												11		5	R
194.												11	20	8	A
195.												14		2	R
196.												14	21	4	R _t =VT
197.												11	21	2	R _t =VT
198.												9	21	5	R _t =VT
199.												6	21	6	R _t =VT

200.												4	19	3	A
201.												6		1	R
202.												6		3	R
203.												6	21	6	R _t =VT
204.												4		1	R
205.												4	21	3	R _t =VT
206.												2		1	R
207.												3	20	1	A
208.												5	20	3	A
209.												7		1	R
210.												7	21	3	R _t =VT
211.												5	21	1	R _t =VT
212.												3	21	3	R _t =VT
213.												1	16	2	A
214.												2	16	5	A
215.												3		1	R
216.												4	18	1	A
217.												6	18	3	A
218.												8		1	R
219.												8		3	R
220.												8	20	6	A
221.												11		5	R
222.												11	20	8	A
223.												14		2	R
224.												14	21	4	R _t =VT
225.												11	21	2	R _t =VT
226.												9	21	5	R _t =VT
227.												6		1	R
228.												6		3	R
229.												6	21	6	R _t =VT
230.												4	19	3	A
231.												6		1	R
232.												6		3	R
233.												6	21	6	R _t =VT
234.												4		1	R
235.												4	21	3	R _t =VT
236.												2		1	R
237.												3	20	1	A
238.												5	20	3	A
239.												7		1	R
240.												7	21	3	R _t =VT
241.												5	21	1	R _t =VT
242.												3	21	3	R _t =VT
243.												1	18	5	A
244.												2		1	R
245.												3	20	1	A
246.												5	20	3	A
247.												7		1	R
248.												7	21	3	R _t =VT
249.												5	21	1	R _t =VT
250.												3	21	3	R _t =VT
251.												1	20	1	R
252.												2	22	1	R _t >VT

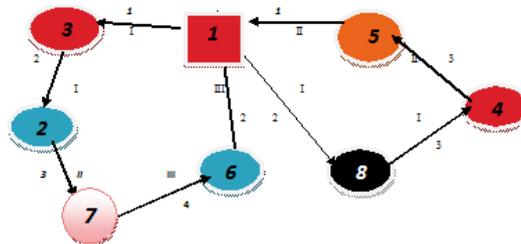
The above table -7 , gives optimal solution for the numerical example and the word is $L_9 = (1,3,5,6,9,11,12,13,16)$ is a feasible word. At the end of search table optimal solution value VT is 21. It is in the 185TH row of the search table. For this word the array IR, IC, SW and L are given in the following Table -8. Then the following figure – 2, represents the optimal solution to the problem Network.

Table -8

	1	2	3	4	5	6	7	8	
L	1	3	5	6	9	11	12	13	16
IR	1+1	1	1	1	1	1	1	1	
IC	1+1	1	1	1	1	1	1	1	
SW	3,8	7	2	5	1	1	6	4	

Optimal Solution:

FIGURE 2



From the above figure-2 ,In tour ,the first sales man starts from city 1 and visit city 3 in first season ,from city 3 he visit citi2 in season I , from city2 he visit city7 in season II , from city 7 he visit city 6 in season III, from city6 he return to city1 in season III. The second sales man starts from city1 and visit city8 in season I, from city8 he visit city 4 in season I, from city 4 he visit city 5 in seasonII, from city 4 he return to city1 in season II . In this way all the cities are covered by two salesman . Hence the value of the solution is as follows

$$Z(X) = D(1,3,1) + D(3,2,1) + D(2,7,2) + D(7,6,3) + D(6,1,3) + D(1,8,1) + D(8,4,1) + (4,5,2) + D(5,1,2) = 1+2+3+4+2+2+3+3+1=21$$

Table-9

X(i, j, 1)							
0	0	1	0	0	0	0	1
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	1	0	0	0	0
0	0	0	0	0	0	0	0

Table-10

X(i, j, 2)							
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	1	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

Table-11

X(i, j, 3)							
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
1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0

The ordered triples are $\{(1,3,1), (5,1,2), (1,8,1), (3,2,1), (6,1,3), (8,4,1), (2,7,2), (4,5,2), (7,6,3)\}$ represents a pattern of X in the tables-9,10&11 is an optimum feasible solution.

V. Conclusion

In this chapter we developed a Lexi-search algorithm to solve **THREE DIMENSIONAL CONSTRAINED P- TSP**. The model is illustrated by a suitable numerical example. Lexi-search algorithm using pattern recognition technique is used to get an optimum solution. The numerical example is tried for better understanding of the concepts and the steps involved in the algorithm. I suggest that this algorithm can perform for larger size problems also.

Acknowledgements:



The first author expresses deep sense of reverence and gratitude to the research supervisor Prof. M .Sundara Murthy, (Rtd), Department of Mathematics, S.V.University, Tirupati, for suggesting this problem for investigation. It is solely due to his immense interest, critical analysis, exceptional guidance, concrete suggestions and enlightened discussions, which cumulatively are responsible for the successful execution of this work.

References

- [1]. Ahuja,R.K.,J.B. Orlin, S. Pallottino, M.G.Scutella. Dynamic shortest paths minimizing travel times and costs. (2003) Networks 41 197-205.
- [2]. Albaich,J, J.M.Sanchis, D.Soler.An asymmetric TSP with time windows and with time dependent travel times and costs: An exact solution through a graph transformation. (2008) European journal of operational research 189 789-802.
- [3]. Bezalel.Gavishat.al:An n-constraint formulation of the time dependent travelling salesman problem.(1980).Operations Research 28 1018-1021.
- [4]. Balakrishna U. [a] Generalized Time Dependent Travelling Salesman Problem with Cyclic constraint [GTSP], [b] Generalized Time Dependent Travelling Salesman Problem (cluster constraint) [GTDTSPP] models, (2009) PhD Thesis, S.V. University, Tirupati.
- [5]. Bhavani. V, Sundara Murthy, M. : Time-Dependent Traveling Salesman Problem,(2005) OPSEARCH 42,199–227
- [6]. Croes, G.A.A Method for Solving Travelling Salesman Problem, (1958)Opns.Res., 6, pp.791-812.
- [7]. Cai,X., T.Kloks, C.K. Wong:Time –varying shortest path problems with constraints. (1997) Networks 29 141-149.
- [8]. Chabini .I:Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimul run time.(1998) Transportation Research Records.1645 170-175.
- [9]. Flood, M. M.: The Travelling Salesman Problem. (1956) Opns.Res. 41, pp.61-75.
- [10]. Fischetti, M., Salazar-Gonzalez, J.J., Toth, P.:A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. (1997) Oper Res 45(3), 378–394
- [11]. Fischetti, M., Salazar, J.J., Toth, P.: The symmetric generalized traveling salesman polytope. (1995) Networks26, 113–123
- [12]. Fomin, F.V, A.Lilngas.Approximation algorithms for the time dependent orienteering.(2002)Information Transportation Science 00 (0), pp. 000-000,c0000 Informs 17.
- [13]. Henry-Labordere, A.: The record balancing problem-A dynamic programming solution of a generalized traveling salesman problem. (1969) Revue Francaise D Informatique De Recherche Operationnelle 3 (NB2), 43–49
- [14]. Laporte, G., Nobert, Y.: Generalized traveling salesman problem through n-sets of nodes-An integer programming approach. INFOR 21(1), 61–75 (1983)
- [15]. Malandraki.C ,R.B.Dial: A restricted dynamic programming heuristic algorithm for the Time dependent travelling salesman problem. European Journal of Operational Research 90 45-55.
- [16]. Miller, C.E at.al :Integer Programming Formulation of TSP. (1960). J. Assoc, Comput. March 7, 326 – 329
- [17]. Montemanni,R., J. Barta, M. Mastrolilli, L.M.Gambardella.The robust travelling salesman problem with interval data(2007). Transportation science 41 366-381.
- [18]. Naganna .B, K.SobhanBabu and M. Sundara Murthy: Continuous Time Dependent Travelling Salesman Problem. (2011) International journal of mathematical research Volume –s, pp 69-78.
- [19]. Orda,A., R.Raphael.Shortest path and minimum-delay algorithms in networks with time–dependent edge length (1990). Journal of the ACM 37 607-625.
- [20]. Picard,J.c., M. Queyranne: The time dependent travelling salesman problem and its application to the tardiness problem in one – machine scheduling. (1978). Operations research 26 86-110.
- [21]. Pandit, S.N.N., Srinivas, K.: A Lexisearch algorithm for traveling Salesman problem,(1962) IEEE 2521–2527
- [22]. Saskena J.P.: Mathematical model of scheduling clients through welfare agencies. (1970),Oper Res Soc8 ,185-200
- [23]. Sundara Murthy, M (1979): Combinatorial Programming - A patternRecognition Approach, Ph.D. Thesis REC, Warangal, India
- [24]. Srivastava, S.S. Santhosh Kumar Gorg, R.C. and Sen. P :GeneralizedTSP through n sets of nodes, (1969): Opns. Res. Vol. 7, No2 pp. 97 - 101
- [25]. SobhanBabu. K, E.Keshava Reddy, M. SundaraMurthy..Travelling Salesman Problem with a grouping Constraint –A Lexi –Search Approach, (2012) International journal of Advanced Research in Computer Science ,Volume-3 pp 463-468.

- [26]. Schneider.J: The time-dependent travelling salesman problem (2002). Physical A: Statistical Mechanics and its Applications 314 151-155.
- [27]. Vander Wiel, R.J., N.V. Sahinidis. An exact solution approach for the time dependent travelling salesman problem. Naval Research Logistics 43 797-820.

Authors:



Y. SREENIVASULU , Research Scholar, Dept., of Mathematics, Sri Venkateswara University, Tirupati



R. VIJAYA LAKSHMI, Research Scholar, Dept., of Mathematics, Sri Venkateswara University, Tirupati



V.K SOMASEKHAR SRINIVAS currently working as Assistant professor in the dept., of Mathematics, Priyadharsini Institute of Engineering and Technology, Tirupati77



Dr. M.SUNDARA MURTHY is a senior Professor in the Department of Mathematics, Sri Venkateswara University, Tirupati.