

P – TSP Seasonal Constrained Model

Vijayalakshmi R¹, Gurava Raju P², Sundara Murthy³

Research scholar, Department of Mathematics, S.V. University, Tirupati, Andhra Pradesh.

rvijayachandra2513@gmail.com

Teacher, Z P H School(Boys),Vadamalpet, Andhra Pradesh

guravarajuc@gmail.com

Professor, Dept., of Mathematics, Sri Venkateswara University, Tirupati, Andhra Pradesh

profmurthy@gmail.com

Abstract: The Multiple Travelling Salesman Problem (MTSP) is a generalization of the well-known TSP, where more than one salesman is allowed to be used in the solution. The characteristics of the MTSP seem more appropriate for real life applications. In this paper we study a problem called P-TSP Seasonal Constrained Model. Let there are number of cities 'n', the number of salesmen P, the third dimension is season S. Let M be common cities, which is subset of n for the salesman. Each salesman has to start their tour from head quarters that is city l in first season only and at the end of the season 1 all the salesmen have to meet at a common city which is in 'M'. Again they starts their tour from common city in next season and visit some more cities and at the end of season 2 each of them have to meet at another common city which is in 'M' like this they travel upto (r-1)th season. Finally in rth season all the salesmen have to reach the head quarter city. All the salesman have to visit 'n₀' cities other than the cities of 'M' that is n₀ < n-m. The objective is to complete p- tours in 'r' seasons with minimum total cost/distance.

Key words: TSP Problem, Lexi-Search, Pattern Recognition Technique, Alphabet Table, Search Table.

INTRODUCTION:

Travelling Salesman Problem is a classical problem in combinatorial optimization. The problem is to plan a tour for travelling salesman to visit 'n' different cities, so that he visits exactly once, returns to his point of origin with least distance. It is and NP-Hard problem in combinatorial optimization.

The method of solving this problem covering the work up to early 1960 is given as a review by Arnoff and Sengupta – 1961. Bellmore and Nemhauser – 1968 made a survey of the work on this problem, covering the period upto 1967. Gupta – 1968 classifies the various methods of solving the TSP as: [a] Combinatorial Approach [b] Network Approach [c] Linear Programming Approach [d] Graphical Approach [f] Heuristic approach.

The TSP has been generalized in many directions. Thus, one such which is called the "Truncated Travelling Salesman Problem" (Sundara Murthy - 1979) is presented as follows:

There are n cities i.e., $N = \{1, 2, \dots, n\}$. The distance $D(i, j)$, between any pair of cities (i, j) is known. A subset of the n cities constitutes the potential places for setting up a head quarters. A salesman has to visit only m out of the n cities is called truncation, with the restriction that his tour should include at least one city from potential headquarters. The problem is to find a feasible tour of m cities with a minimum length.

2.3. PROBLEM DESCRIPTION:

In this chapter we have a variant multiple TSP called P-TSP Seasonal Constrained Model. Let there be a set of 'n' cities, P - salesmen and 'r' seasons.

We consider 'n' number of cities $N = \{1, 2, \dots, n\}$, the number of salesmen $P = \{1, 2, \dots, p\}$. An independent factor which influences the cost is introduced as a third dimension. In this model third dimension is season, let $S = \{1, 2, \dots, r\}$ is the season. Let $M = \{a_1, a_2, \dots, a_m\} \subset N$ be eligible common cities

for the salesman. All the salesman has to start their tour from head quarter city 1 in first season only, they visit few cities and at the end of a season 1 all the salesmen has to meet at a particular (same) common city which is in 'M'. Again they starts their tour from that common city into next season (2nd season) and visit some more cities and at the end of season 2 each of them has to meet at another particular city which is in 'M' like this they travel upto (r-1)th season. Finally in rth season all the salesmen have to reach at head quarter city.

All the salesman together have to visit 'n₀' cities other than 'm' that is n₀< n-m. The distance array D(i, j, k) indicates the distance / cost of a salesman by visiting the jth city from ith city in rth season.

The objective is to find p- tours in 'r' seasons with minimum total cost/distance, subject to conditions.

2.4. MATHEMATICAL FORMULATION:

$$\text{MIN } Z = \sum_{i \in N} \sum_{j \in N} \sum_{k \in S} D(i, j, k) x(i, j, k) \dots (1)$$

Subject to constraints:

$$\sum_{i \in N} x(i, \alpha_k, k) = p, k=1,2,\dots,r \text{ \& } \alpha_k \in M \dots (2)$$

$$\sum_{i \in N} x(\alpha_k, i, k+1) = p \text{ if } k=r, k+1=1 \dots (3)$$

$$\sum_{i \in N} \sum_{j \in N} \sum_{r \in S} x(i, j, k) = n_0 + r + r \dots (4)$$

$$x(i, j, r) = \begin{cases} 0 \\ 1 \end{cases} \text{ for } i \in N, j \in N \text{ \& } r \in S \dots (5)$$

Equation (1) represents the objective of the problem, that is the total minimum distance travelled by P salesmen.

Constraint (2) denotes total number of tours

Constraint (3) states at the end of the kth season all the 'p' salesmen come through ith city to α_kth city.

Constraint (4) indicates that the total number of connectivities.

Constraint (5) illustrates if ith salesman visit to jth city during rth season then the value is 1 other wise 0

2.5. NUMERICAL ILLUSTRATION:

We consider a numerical example with 11 cities, 2 salesmen with 3 seasons (matrix of order 11*11*3) and solved the problem by lexicographic search.

The concepts and the algorithm developed will be illustrated by a numerical example. For this problem we considered total number of cities N={1,2,...,11}, number of salesmen P={1,2}, the number of seasons S={1,2,3} and common cities are denoted as M={1,4, 8}, the two salesmen has to finish their tour by visiting n₀=8 cities in 3 seasons where n=11. The distance matrices according to seasons are given in **table -1** as follows

TABLE-1

	1	2	3	4	5	6	7	8	9	10	11
1	8	16	--	20	2	40	--	24	--	12	1
2	19	8	32	--	43	11	51	--	60	--	29
3	--	62	8	36	--	47	30	35	08	63	--
4	39	--	25	8	67	17	--	61	--	53	41
5	2	35	--	4	8	54	9	--	45	--	38
6	--	44	27	--	66	8	39	--	33	18	--
7	65	--	71	--	15	21	8	--	50	--	56
8	--	52	10	42	--	58	31	8	73	--	26
9	69	--	48	14	72	--	22	57	8	74	--
10	--	34	--	75	46	--	70	--	76	8	28
11	23	--	64	3	--	49	--	37	--	5	8

	1	2	3	4	5	6	7	8	9	10	11
1	8	65	--	34	22	--	63	73	46	51	--
2	20	8	74	--	71	41	--	12	17	--	38
3	70	52	8	60	10	--	29	36	--	55	75
4	--	18	62	8	16	2	--	39	--	3	64
5	67	14	--	35	8	27	11	--	48	19	--
6	--	40	25	--	33	8	50	1	--	9	--
7	44	--	8	54	--	21	8	45	--	24	37
8	--	28	30	--	42	15	--	8	26	--	--
9	72	--	47	43	--	--	66	4	8	58	--
10	--	33	--	4	31	36	--	5	6	8	69
11	23	--	57	68	--	49	61	--	59	32	8

	1	2	3	4	5	6	7	8	9	10	11
1	8	37	--	19	55	21	--	40	1	--	32
2	2	8	27	--	76	31	15	--	75	--	54
3	3	65	8	39	9	--	45	17	--	64	--
4	24	--	12	8	53	14	--	33	--	28	20
5	--	47	30	66	8	56	25	11	44	--	72
6	18	52	--	8	63	8	16	--	29	57	--
7	--	67	5	--	22	48	8	70	--	34	74
8	35	4	71	26	--	43	4	8	4	--	38
9	51	4	68	--	49	--	61	44	8	58	--
10	46	--	43	--	36	50	--	--	5	8	60
11	--	64	--	62	10	--	23	3	59	42	8

In cost matrices C(i,j,1), C(i,j,2) & C(i,j,3), i, j ∈ N, if i=j the values are taken as ∞ because they are irrelevant of finding tours for the salesmen and '–' are dis connectivity.

All the C(i,j,k) are taken as positive integers but it can be easily seen that this is not a necessary condition and the distance/cost can as well real quantities. For example C(5,4,1)=4 represents the distance/cost of the travelling salesman by visiting 4th city from 5th city in 1st season is 4 units.

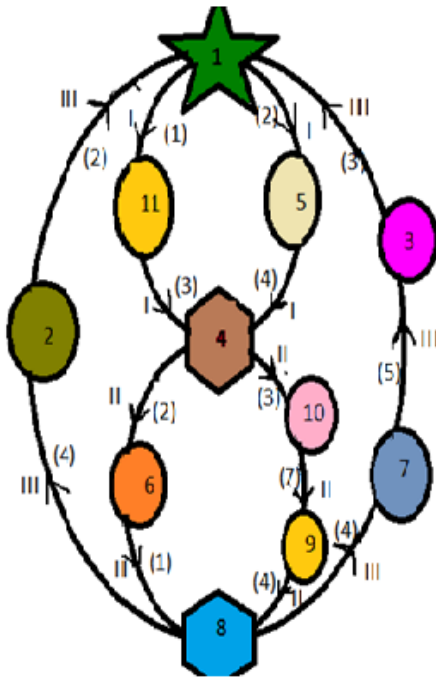
2.6 FEASIBLE TRIP:

Consider an ordered triple set {(1, 11, 1), (6, 8, 2), (1, 5, 1), (4, 6, 2), (2, 1, 3), (11, 4, 1), (4, 10, 2), (3, 1, 3), (5, 4, 1), (9,

8, 2), (8, 2, 3), (8, 7, 3), (7, 3, 3), (10, 9, 2)} represents which is a feasible solution set

In the below **figure-1**, Circles represents the cities, Star represents the Headquarter city {1}, Hexagon represents the common cities, the values within the parenthesis represents the distance/cost of the travelled between two cities, the roman numerical represents seasons.

Figure- 1 [FEASIBLE SOLUTION]



In the above feasible solution first sales man starts his tour from headquarter city (1) in first season. He visits from city 5 from city 1 at distance/cost 2 units, from city 5 he travel to city 4 which is a common city with a distance 4 units. Again he starts from common city (4) in second season, he visits city 10 from city 4 with a distance of 3 units, from city 10 he travel to city 9 with a distance 7 units and from city 9 he travel to city 8 which another common city with a distance 4 units. Again he starts his tour from second common city 8 in third season, he visits city 7 from city 8 with a distance of 4 units, from city 7 he travel to city 3 with a distance of 5 units and from city 3 he travel to city 1 (headquarter) with a distance of 3 units.

Second salesmen starts his from headquarter city (1) in first season. He visits city 11 from city 1 at a distance/cost 1 unit, from city 11 he travel to city 4 which is a common city with a distance 3 units. Again he starts from common city (4) in second season, he visits city 6 from city 4 with a distance 2

units, from city 6 he travel to city 8 which another common city with a distance 1 unit. Again he starts tour from second common city 8 in third season, he visits city 2 from city 8 with a distance 4 units, from city 2 he travel to city 1(headquarter) with a distance of 2 units.

2.7. SOLUTION PROCEDURE:

In the above **figure-1**, for the feasible solution we observed that there are 14 ordered triples {(1, 11, 1), (6, 8, 2), (1, 5, 1), (4, 6, 2), (2, 1, 3), (11, 4, 1), (4, 10, 2), (3, 1, 3), (5, 4, 1), (9, 8, 2), (8, 2, 3), (8, 7, 3), (7, 3, 3), (10, 9, 2)} taken along with the value from the cost matrices in the numerical example in **table-1**. The 14 ordered triples are selected such that they represents a feasible solution according to constraints of mathematical formulation and is represented in **figure-1**. So the problem is that we have to select fourteen ordered triples from the cost matrices along with values such that the total cost is minimum and represents a feasible solution. For this selection of fourteen ordered triples from cost matrices we arranged ordered triples with the increasing order of their values and call this formation as alphabet table and we will develop an algorithm for the selection of **fourteen ordered triples** along with the checking for the feasibility.

2.8. CONCEPTS AND DEFINITIONS:

A tour is a feasible trip - schedule for the salesman. Trip-schedule of the salesman can be represented by an approximate $n \times n \times r$ indicator array $X = \{x(i, j, k); x(i, j, k) = 0 \text{ or } 1\}$ in which $x(i, j, k) = 1$ indicates that the salesman visits city j from city i in k^{th} season.

DEFINITION OF A PATTERN:

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

$$V(x) = \sum_{i \in N} \sum_{j \in N} \sum_{k \in S} D(i, j, k). x(i, j, k)$$

The value $V(x)$ gives the total cost/distance of the tour for the solution represented by X . The pattern represented in the table-2 is a feasible pattern. The value $V(X)$ gives the total cost of the tour for the solution represented by X . Thus the value of the feasible pattern gives the total cost represented

by it. In the algorithm, which is developed in the sequel, a search is made for a feasible pattern with the least value. 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 $X(i, j, k)$'s are zeros.

Table – 2

$$\begin{aligned}
 & X(i,j,1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 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 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ = & \begin{bmatrix} 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 & 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 & 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 \end{bmatrix} \\
 \\
 & X(i,j,2) = \begin{bmatrix} 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 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ = & \begin{bmatrix} 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 & 1 & 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} \\
 \\
 & X(i,j,3) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ = & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 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 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

The above pattern represents a feasible pattern, the ordered triples and the corresponding distance/cost values according to pattern is given as follows

$$(1,5,1)+(1,11,1)+(5,4,1)+(11,4,1)+(4,6,2)+(4,10,2)+(6,8,2)+(9,8,2)+(10,9,2)+(2,1,3)+(3,1,3)+(7,3,3)+(8,2,3)+(8,7,3)=2+1+4+3+2+3+1+4+7+2+3+5+4+4=45 \text{ units.}$$

2.8.2 DEFINITION OF AN ALPHABET - TABLE AND A WORD:

There are $N= n*n*r$ ordered triples in the three-dimensional array C. For convenience these are arranged in ascending order of their corresponding cost and are indexed from 1 to $n*n*r$ (Sundara Murthy-1979). Let D be the corresponding array of cost. If $a,b \in SN$ and $a < b$ then $D(a) \leq D(b)$. Let the arrays R, C, K be the row, column and season indices of the ordered triples represented by S.No and DC be the array of cumulative sum of the elements of D. The arrays S.No, D, DC, R, C, K for the numerical example are given in the table-3. If $p \in S.no$ then $(R(p), C(p), K(p))$ is the ordered triple and $D(a) = D(R(a), C(a), K(a))$ is the value of the ordered triple and $DC(a) = \sum_{i=1}^a D(i)$ for convinence 'C' in array represented in column.

ALPHABET TABLE - 3

S.No	D	DC	R	C	K
1	1	1	1	11	1
2	1	2	6	8	2
3	1	3	1	9	3
4	2	5	1	5	1
5	2	7	5	1	1
6	2	9	4	6	2
7	2	11	2	1	3
8	3	14	11	4	1
9	3	17	4	10	2
10	3	20	3	1	3
11	3	23	11	8	3
12	4	27	5	4	1
13	4	31	9	8	2
14	4	35	8	2	3
15	4	39	8	7	3
16	4	43	8	9	3
17	4	47	9	2	3
18	5	52	10	8	2
19	5	57	7	3	3
20	7	64	10	9	2
21	8	72	3	9	1
22	8	80	7	3	2
23	8	88	6	4	3
24	9	97	5	7	1
25	9	106	6	10	2
26	9	115	3	5	3
27	10	125	8	3	1
28	10	135	3	5	2
29	10	145	11	5	3
30	11	156	2	6	1

Let $S.No = \{1,2,\dots\}$ be the set of indices, D be an array of corresponding costs of the ordered triples and DC be the array of cumulative sum of elements in D . Let arrays R, C and K be respectively, the row, column and season indices of the ordered triples. Let $L_k = \{a_1, a_2, \dots, a_k\}$, $a_i \in S.No$ be an ordered sequence of k indices from $S.No$. 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 $S.No$ is defined as the "Alphabet-Table" with alphabetic order as $(1, 2, \dots, n*n*r$ and the ordered sequence L_k is defined as a "word" of length k . A word L_k is called a "sensible word". If $a_i < a_{i+1}$, for $i = 1, 2, \dots, k-1$ and if this condition is not met it is called a "insensible word". A word L_k is said to be feasible if the corresponding pattern X is feasible and same is with the case of infeasible. A Partial word L_k is said to be feasible if the block of words represented by L_k has at least one feasible word or, equivalently the partial pattern represented by L_k should not have any inconsistency. Any of the letters in $S.No$ can occupy the first place in the partial word L_k . Our interest is only in set of words of length at most equation, since the words of length greater than n are necessarily infeasible, as any feasible pattern can have only n unit entries in it. If $k < n$, L_k is called a partial word and if $k = n$, it is a full length word or simply a word. A partial word L_k represents, a block of words with L_k as a leader i.e. as its first k letters. A leader is said to be feasible, if the block of word, defined by it has at least one feasible word.

2.8.3 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 $V(L_0) = 0$ where $D(a_k)$ is the cost array arranged such that $D(a_k) < D(a_{k+1})$. Since X is the (partial) pattern is represented by L_k , (Sundara Murthy – 1979).

For example the word $L_4 = (1, 2, 4, 6)$ then value of L_4 is

$$\begin{aligned} V(L_4) &= V(L_{4-1}) + D(a_4) \\ &= V(L_3) + D(a_4) \\ &= 4 + 2 = 6 \end{aligned}$$

2.8.3 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 can be defined as follows:

$$LB(L_k) = V(L_k) + DC(a_k + m - k) - DC(a_k)$$

Consider the partial word $L_4 = (1, 2, 4, 6)$, here $a_k = 6$, $m = 14$ and $k = 4$

$$\begin{aligned} LB(L_4) &= V(L_4) + DC(6 + 14 - 4) - DC(6) \\ &= 6 + DC(16) - DC(6) \\ &= 6 + 43 - 9 \\ &= 40 \end{aligned}$$

FEASIBILITY CRITERION OF A PARTIAL WORD:

An algorithm was developed, in order to check the feasibility of a partial word $L_{k+1} = \{a_1, a_2, \dots, a_k, a_{k+1}\}$ 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 $IR(i) = 1$, $i \in N$ indicates that the sales man is visits some city from city i Otherwise $IR(i) = 0$
- **IC** be an array where $IC(j) = 1$, $i \in N$ indicates that the sales man is coming to city i from another city, otherwise $IC(j) = 0$
- **IK** be an array where $IK(i) = 1$, $i \in S$ represents that the salesman at season (facility) i travels one pair of cities.
- **SW** be an array where $SW(i) = j$ indicates that the sales man is visiting city j from city i , Otherwise $SW(i) = 0$
- **SWI** be an array where $SWI(j) = i$ indicates that the salesman is visiting city i from city j , otherwise $SWI(j) = 0$.
- **L** be an array where $L(i) = \alpha_i$, $i \in N$ is the letter in the i^{th} position of a word.

Then for a given partial word $L_k = (a_1, a_2, \dots, a_k)$ the values of the arrays IR, IC, IK, SW, SWI and L as follows.

$$\begin{aligned} IR(R(a_i)) &= 1, & i &= 1, 2, 3, \dots, K \\ IC(C(a_i)) &= 1, & i &= 1, 2, 3, \dots, K \\ IK(K(a_i)) &= 1, & i &= 1, 2, 3, \dots, K \\ SW(R(a_i)) &= C(a_i), & i &= 1, 2, 3, \dots, K \\ SWI(C(a_i)) &= R(a_i) \\ L(i) &= a_i, & i &= 1, 2, 3, \dots, K \end{aligned}$$

For example consider a sensible partial word $L_7 = \{1, 2, 4, 6, 7, 8, 9\}$ which is feasible. The array $IR, IC, IK, SW,$ and L takes the values represented in table – 4 given below.

TABLE – 4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	1	2	4	6	7	8	9							
IR	2	1	2		1									
IC	1			1	1	1		1	--	1	1			
IK	1	2	1	2	3	1	2							
SW	11,5	1	10,6	8										
SWI	2,		11,	1	4			6		4	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 $RA = R(a_{p+1})$, $CA = C(a_{p+1})$ and $KA = K(a_{p+1})$.

ALGORITHM-1 : (Feasible checking) :

Step 101 : $IX=0$ go to 102
 Step 102 : $MX(RA)=1$ yes go to 104 , no go to 122
 Step 104 : Is $RA=1$ yes go to 106 , no go to 112
 Step 106 : Is $KA=1$ yes go to 108, no go to 144
 Step 108 : Is $SW(CA)=0$ yes go to 142, no go to 110
 Step 110 : Is $ST(CA)=KA$ yes go to 110a, no go to 144
 Step 110a : Is $SWI(RA)=0$ yes go to 142, no go to 112
 Step 112 : Is $STC(RA)=KA$ yes go to 114, no go to 144
 Step 114 : Is $SW(CA)=0$ yes go to 142, no go to 115
 Step 115 : Is $MX(CA)=1$ yes go to 116, no go to 118
 Step 116 : Is $ST(CA)=KA$ yes go to 142, no go to 144
 Step 118 : Is $KA=ST(CA)-1$ yes go to 142, no go to 144
 Step 122 : Is $MX(CA)=1$ yes go to 124, no go to 134
 Step 124 : Is $(CA)=1$ yes go to 126, no go to 129
 Step 126 : Is $KA=r$ yes go to 128, no go to 144
 Step 128 : Is $KA=ST(SWI(RA))$ yes go to 142,
 no go to 144
 Step 129 : Is $ST(CA)=0$ yes go to 132, no go to 130
 Step 130 : Is $ST(RA)=ST(CA)-1$ yes go to 132,
 no go to 144
 Step 132 : Is $ST(SWI(RA))=KA$ yes go to 134,
 no go to 144
 Step 134 : Is $(SW(CA))=0$ yes go to 138, no go to 136
 Step 136 : Is $ST(SW(CA))=KA$ yes go to 138,

no go to 144

Step 138 : Is $SWI(RA)=0$ yes go to 142, no go to 140

Step 140 : Is $ST(SWI(RA))=KA$ yes go to 142,

no go to 144

Step 142 : $IX=1$

Step 144 : END

ALGORITHM-II : LEXI-SEARCH ALGORITHM

Step 1 : (Initialization)

The arrays SN, D, CD, R, C, K, Max and B are made available MX, IR, IC, ST, L, V, LB, SW, SWI, STC, RX, SW(1,i), SW(2,i), SWI(1,i), SWI(2,i) are initialized to zero. The values $I=1, J=0, VT=999, m=3, n=11, RA=0, CA=0, KA=0,$

Step 2 : $J=J+1$

Is $(J \geq \text{Max})$ yes go to 11 , no go to 3

Step 3 : $L(I)=J$

$RA=R(J)$

$CA=C(J)$

$KA=K(J)$

go to 4

Step 4 : $V(I)=V(I-1)+D(J)$

$LB(I)=V(I)+CD(J+n+r-I)-CD(J)$ go to 5

Step 5 : Is $LB(I) \geq VT$ yes go to 11 , no go to 6

Step 6 : [Check the feasibility of using algorithm-1]

Is $IX=1$ yes go to 8a , no go to 2

Step 7 : Is $I=n+m$ yes go to 9 , no go to 8

Step 8 : $I=I+1$

go to 2

Step 8a : $IR(RA)=IR(RA)+1;$

$IC(CA)=IC(CA)+1;$

$MX(CA)=1;$ yes go to 8b, no $ST(RA)=KA$

Step 8b : $ST(CA)=KA +1$

go to 8c

Step 8c : $MX(RA) \neq 1;$

$MX(CA) \neq 1;$ yes go to 8d, no $SW(RA)=CA;$

$SWI(CA)=RA$

Step 8d : $IR(RA)=RX$

go to 8e

Step 8e : $MX(RA)=1$

yes $SW(RX,RA)=CA;$

$SWI(RX,CA)=RA$, no goto 2

Step 9 : $VT=V(I)$

$L(I)=J$, record VT

go to 10

TABLE – 5

Step 10 : I= I-1;

J = L(I);

RA = R(J);

CA = C(J);

go to 10a

Step 10a : IR(RA)= IR(RA)-1;

IC(CA)= IC(CA)-1;

go to 10b

Step 10b : IR(RA)= RX;

IC(CA)= CX;

go to 10c

Step 10c : MX(RA)≠1;

MX(CA) ≠1; yes go to 10d, no go to 10e

Step 10d : SW(RA)=0;

SWI(CA)=0;

ST(RA)=0;

go to 2

Step 10e : MX(RA)=1; yes go to 10f, no go to 10

Step 10f : SW(RA)=0;

SW((RX+1),RA)=0;

SWI((RX+1),CA)=0; goto 2

Step 11 : IS(I=1) yes go to 12 , no go to 10

Step 12 : STOP

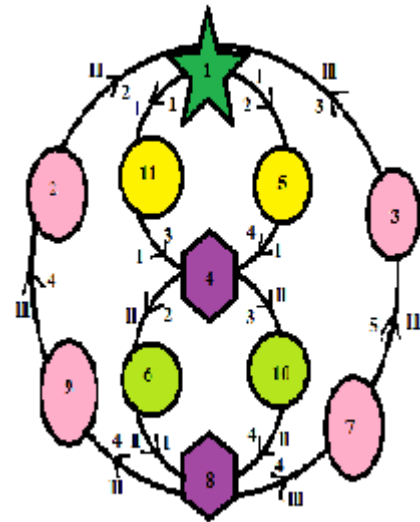
SEARCH TABLE:

The working details of getting an optimal word using the above algorithm for the illustrative numerical example is given in the Table – 5. 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 season indices of the letter. The last column gives the remarks regarding the acceptability of the partial words. In the following table A indicates ACCEPT and R indicates REJECT.

SEARCH TABLE

S.No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	V	IB	R	C	K	REMARK		
1	1														1	35	1	11	1	A		
2		2													2	35	6	8	2	A		
3			3												3	35	1	6	3	R		
4				4											4	38	1	5	1	A		
5					5										5	38	5	1	1	R		
6						6									6	40	4	6	2	A		
7							7								7	40	2	1	3	A		
8								8							8	41	11	4	1	A		
9									9						9	40	4	10	2	A		
10										10					10	40	3	1	3	A		
11											11				11	40	11	8	3	R		
12												12			12	41	5	4	1	A		
13													13		13	41	9	6	2	A		
14														14	14	41	8	2	3	A		
15															15	33	41	8	7	3	A	
16															16	51	41	8	5	3	R	
17															17	37	42	9	2	3	R	
18															18	38	43	10	6	2	R	
19															19	38	43	7	3	3	A	
20															20	45	45	10	9	2	A, V1=45	
21															21	40	48	10	9	2	R, >VT	
22															22	33	47	8	6	3	R	
23															23	33	43	9	2	3	R	
24															24	34	46	10	8	2	R, >VT	
25															25	29	42	8	7	5	A	
26															26	33	42	8	9	3	R	
27															27	33	43	9	2	3	R	
28															28	33	45	10	6	2	R, >VT	
29															29	29	43	8	5	3	R	
30															30	29	46	9	2	3	R, >VT	
31															31	25	42	8	2	3	A	
32															32	29	47	8	7	3	A	
33															33	33	42	8	9	3	R	
34															34	17	53	45	9	2	5	R
35															35	18	31	46	10	8	2	R, >VT
36															36	29	43	8	9	3	R	
37															37	17	29	46	9	2	3	R, >VT
38															38	15	25	43	8	7	3	A
39															39	15	29	43	8	5	3	A
40															40	17	43	4	7	4	A	
41															41	13	38	43	10	6	2	A
42															42	19	43	43	7	3	3	A, V1=43
43															43	19	38	45	7	3	3	R, >VT
44															44	18	34	46	10	6	2	R, >VT
45															45	17	29	46	9	2	3	R, >VT
46															46	16	25	46	8	9	3	R, >VT
47															47	15	21	42	9	8	2	A
48															48	14	25	42	8	2	3	A
49															49	15	25	42	8	7	3	A
50															50	16	33	42	8	9	3	R
51															51	17	33	42	9	2	3	R, >VT
52															52	16	25	43	8	9	3	R, >VT
53															53	15	25	43	8	7	3	R, >VT
54															54	14	21	43	8	2	3	R, >VT
55															55	11	17	41	11	8	3	R

56				12				18	43	5	4	.	R,=VT
57				10				14	43	3	1	.	A
58				11				17	43	11	3	.	R
59				12				18	43	5	4	.	R,=VT
60				11				14	43	11	3	.	R,=VT
61			9					11	41	4	10	2	A
62				10				14	41	3	1	3	A
63				11				17	41	11	3	3	R
64				12				18	43	5	4	.	R,=VT
65				11				14	43	11	3	.	R,=VT
66				10				11	43	3	1	3	A
67			8					09	42	11	4	.	A
68				9				12	42	4	10	2	A
69				10				15	42	3	1	3	A
70				11				18	42	11	3	3	R
71				12				19	44	5	4	.	R,=VT
72				11				15	44	11	3	3	R,>VT
73				10				12	44	3	1	3	R,>VT
74			9					09	44	4	10	2	R,>VT
75			7					06	42	2	1	3	A
76				8				09	42	11	4	.	A
77				9				12	42	4	10	2	A
78				10				15	42	3	1	3	A
79				11				18	42	11	3	3	R
80				12				19	44	5	4	.	R,>VT
81				11				15	44	11	3	3	R,>VT
82				10				12	44	3	1	3	R,>VT
83				9				09	44	4	10	2	R,>VT
84			8					06	44	11	4	.	R,>VT
85			5					04	40	5	1	.	R
86			5					04	42	4	5	2	A
87			7					06	42	2	1	3	A
88				8				09	43	11	4	.	A
89				9				12	42	4	10	2	A
90				10				15	42	3	1	3	A
91				11				18	42	11	3	3	R
92				12				19	44	5	4	.	R,>VT
93				11				15	44	11	3	3	R,>VT
94				10				12	44	3	1	3	R,>VT
95				9				09	44	4	10	2	R,>VT
96			8					07	45	11	4	1	R,>VT
97				7				04	45	2	1	3	R,>VT
98		3						02	38	1	9	3	R
99		4						03	41	1	5	1	A
100			5					05	41	5	1	1	R
101			6					05	43	4	6	2	R,=VT
102			5					03	43	5	1	1	R,=VT
103	2							01	38	5	8	2	A
104		3						02	38	1	9	3	R
105		4						03	41	1	5	1	A
106			5					05	41	5	1	1	R
107			6					05	43	4	6	2	R,=VT
108			5					03	43	5	1	1	R,=VT
109	3							01	41	1	9	3	R
110	4							02	44	1	5	1	R,>VT



In the above optimal solution first salesman starts his tour from headquarter city 1 in first season. He visits city 5 from city 1 at distance/cost 2 units, from city 5 he travels to city 4 which is a common city with a distance 4 units. Again he starts from common city 4 in second season, he visits city 10 from city 4 with a distance a of 3 units, from city 10 he travel to city 8 which another common city with a distance 4 units. Again he starts his tour from second common city 8 in third season, he visits city 7 from city 8 with a distance of 4 units, from city 7 he travel to city 3 with a distance of 5 units and from city 3 he travel to city 1(headquarter) with distance of 3 units.

Second salesmen starts his from headquarter city 1 in first season. He visits city 11 from city 1 at distance/cost 1 unit, from city 11 he travel to city 4 which is a common city with distance 3 units. Again he starts from common city 4 in second season, he visits city 6 from city 4 with a distance of 2 units, from city 6 he travel to city 8 which another common city with a distance 1 unit. Again he starts tour from second common city 8 in third season, he visits city 9 from city 8 with a distance of 4 units, from city 9 he travel to city 2 with a distance of 4 units, from city 2 he travel to city 1(headquarter) with a distance of 2 units.

The array IR, IC, IS, SW, SWI and L takes the values represented in the Table-6 given below.

Table – 6

At the end of the search the current value of VT is 43 and it is the value of the optimal feasible word $L_{14} = (1, 2, 4, 6, 7, 8, 9, 10, 12, 15, 16, 17, 18, 19)$. It is given in the 42nd row of the search table and the corresponding order triples are (1,5,1), (1,11,1), (5,4,1), (11,4,1), (4,6,2), (4,10,2), (6,8,2), (10,8,2), (2,1,3), (3,1,3), (7,3,3), (8,9,3), (9,2,3), (8,7,3) Then the following figure – 2 represents the optimal solution .

Figure- 2 [OPTIMAL SOLUTION]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	1	2	4	6	7	3	9	10	12	15	15	17	18	19
IR	2	1	1	2	1	1	1	2	1	1	1			
IC	2	1	1	2	1	1	1	2	1	1	1			
IK	1	3	3	2	1	2	3	3	2	2	1			
SW	11,5	1	1	10,5	4	3	3	9,7	2	8	4			
SWI	2,3	5	7	11,5	1	4	8	10,6	8	4	1			

The Pattern represented by the above optimal feasible word is given below in the following table-7.

Table – 7

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

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

$$X(i,j,3) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The above pattern represents an optimal pattern, the ordered triples and the corresponding distance/cost values according to pattern is given as follows

$$(1,5,1)+(1,11,1)+(5,4,1)+(11,4,1)+(4,6,2)+(4,10,2)+(6,8,2)+(10,8,2)+(2,1,3)+(3,1,3)+(7,3,3)+(8,9,3)+(9,2,3)+(8,7,3)=2+1+4+3+2+3+1+4+2+4+4+4+4+4+5=43 \text{ units.}$$

CONCLUSION:

In this chapter, we have studied a model of travelling salesman problem called P-TSP Seasonal Constrained Model. We developed a Lexi-Search Algorithm using Pattern Recognition Technique for getting an optimal solution. First the model is formulated into a zero-one programming problem.

REFERENCES:

1. ARNOFF, E. L AND SENGUPTA, S.S. The TSP progress in OR., *Willey, New York*, 1 (1961) 150-157.
2. BELLMORE, M. AND NEMHAUSER, G. The TSP-A Survey, *Ops. Res.* 16, , (1968) 538–558.
3. GUPTA, J. N. D. Combinatorial Search Algorithm for a scheduling problem. M.Tech Thesis, IIT, Kharagpur, 1967.
4. GUPTA, J. N. D. TSP - A Survey of Theoretical Developments and applications. *Opsearch*, 5(4) (1968) 181-192.
5. ANEJA, Y.P. AND NAPIAR K.P.K. BICRITERIA transportation problems, *Management Science*, 25 (1979) 73-78.
6. BALACHANDRAN, V. An Integer Generalized Transportation model for optimal Job Assignment in *Computer Networks*, 24 (1976) 742 – 759.
7. BALAKRISHNA, U. Operations Research, Ph. D thesis, S. V. University, Tirupati, India 2009.
8. BHAVANI, V. & SUNDARA MURTHY M. Truncated M-Travelling Salesmen Problem, *OPSEARCH*, 43(2) 2006.
9. BHAVANI, V. Combinatorial Programming Problems (TSP) M.Phil Thesis, *SVU College of Engg*, Tirupati, A.P., India, 1997.

10. CHABINI, I. Discrete dynamic shortest path problems in transportation applications, *Transportation Research Record* 1645, 1998.
11. CHANNAS, S., KOLODZIEJCZYK, W., AND MACHAJ, A. A FUZZY approach to the transportation problem, *Fuzzy Sets and Systems*, 13(1984) 211-221.
12. CHRISTOFIDES, N AND ELION.S. Algorithm for large scale TSP. *Opns. Res. Qlty*, 23(1972) 511 – 518.
13. CRORES, G. A. "A method for solving TSP" *Opns. Res.6* (3) (1958)791.
14. DAS. S. "The Most Economical Route for a TSP" paper presented in ORSI. 1978.
15. LITTLE, J.D.C. MURTHY K.G. SWEENY, D.W. An algorithm for the TSP. *Opns. Res.*, 11 (1963) 972-982.
16. NAGANNA B. Operations Research. Ph. D thesis, S. V. University, Tirupati, India. (2007),
17. NAGANNA, B., SOBHAN BABU, K., and SUNDARA MURTHY M., Continuous Time dependent Travelling Salesman Problem, *International Journal of Mathematics Research*, 3(1) (2011) 69-78.
18. PANDIT, S.N.N. and RAJBHOUGSHI. Restricted TSP through n sets of nodes. *Paper presented at the 9th Annual Convention of ORSI*, Calcutta, 1976.
19. SHILA DAS, NAZIMUDDIN AHMED, A Travelling Salesman Problem (TSP) with Multiple Job Facilities published, *opsearch*, 38(4) (2001)2595-2600.
20. SOBHAN BABU.K., CHANDRA KALA. K., NAGANNA, B and SUNDARA MURTHY. M, A Lexi-Search Approach for Variant Multiple Travelling Salesmen Problem, *International journal of advanced sciences in computer sciences* 1(3) (2010) 169-172.
21. SUNDARA MURTHY, M AND BHAVANI. V – 2006. Truncated M-TSP, *Opsearch*, 43(2)152-177.