Lexicographic optimization of Travelling Salesman Problem with multiple Job facilities and Precedence Constraints

Dr. Krishan Murari Agrawal
Associate Professor, Mathematics, Bipin Bihari (P.G) College, Jhansi (up)-India

***Abstract* - Lexicographic Approach is used to solve a Travelling salesman problem with multiple Job facilities and Precedence Constraints. As multiple jobs are at various station so some stations may be left unvisited, so modified Cost matrix is made Then, according to the Lexicographic algorithm developed in this paper is followed, The Lexicographic algorithm thus developed is tested on MATLAB software also.**

***Index Terms* - lexicographic approach, precedence constraints, travelling salesman problem, tour.**

1.INTRODUCTION

Travelling salesman problem is of great interest in past 70 years. Many techniques has been made by researchers to solve TSP in various situations. TSP is that problem in which a salesman is to visit various cities, Starting from home station and coming back to home station back. He has to choose path in such a way the travelling cost, travelling time etc are consumed minimum after completing all the jobs done.

Affenzeller M. et al.2003[1] solved TSP with Genetic algorithm ,Bianco et al 1994 [2] solved TSP by Dynamic Programming, Carpaneto G.1995[4] by Branch and Bound algorithm. Here in this paper our main interest is in Lexicographic search. Section 2 of this paper contains definition of Lexicographic search. Section 3 has Mathematical formulation of the Travelling salesman problem ,Definitions of the terms used in the Algorithm developed in the paper and The Lexicographic Algorithm itself. Section 4 is Numerical Illustration and Section 5 is conclusion.

2. LEXICOGRAPHIC METHOD

Let a, b, c, d are the name of the stations on which the salesman is to visit. Lexicographic method works as Dictionary method. when a salesman starts from 'a' it can go 'b', 'c', 'd' so the route becomes 'ab', 'ac', 'ad'. Let 'ab' is the selected route then next route may be 'abc' or 'abd'. Now let 'abc' is the route then 'abcd' is the only way. Rest can be understood by the above figure.

A network with n stations with 'T' as headquarter and a cost matrix [Distance or Time matrix] D= $[d_{ij}]_{nxn}$ associated with ordered station pairs (i, j) is given. A set $[J_1 J_2....J_m]$ of m jobs, have been given according to their availability at stations. Also a set of k precedence relations between station pairs ($i_r < j_r$: r = 1,2, -----s) representing the restrictions that any cycle (tour) is to satisfy for it's acceptability, is given.

A salesman starts from station 'i' (Head quarter) and returns to it after completing all the jobs. He completes all the m jobs either by visiting all the n stations or a subset of it and each station only once. And he should not visit the station already visited. No job available at home station. The mathematical formation of the problem is given in the following 3.1(a)

3.1 (a) Mathematical formulation

Mathematically the problem may be stated as

Minimize Z = $\sum_i \sum_j d$ (A_i, A_j) x (A_i, A_j);i. j=1,2,, (n-1) = $\sum_i \sum_j d_{ij}\, x_{ij}$

(For simplicity we write d $(A_i, A_j) = d_{ij}$ and x $(A_i, A_j) = x_{ij}$) subject to

$\sum_{j=1}^{n-1} x_{ij} = 1,\ \sum_{j=1}^{n-1} x_{ji} = 1,$

(Since the salesman starts from a depot say 1 and goes back to it)

$\sum_{j \neq k} x_{jk} = \sum_{k \neq j} x_{kj} = 1$,subjected to $i_r < j_r$: r = 1,2, -s

The unwanted sub-tours are eliminated by lexicographic search procedure.

3.2 THE SOLUTION PROCEDURE :

3.2 (a) Formation of modified matrix

We prepare a check list to find out which station must be visited and which station may be ignored, so that all jobs may be completed, by minimum journey.

(i) We frame a table by denoting the columns as 'jobs' and rows as 'stations'.

(ii) Write J in the cell where the jobs (J_1 J_2....J_m) that are available at the corresponding station (1, 2,. .n).

(iii) We mark Δ to the J, that are single in that column, and mark Δ to the other J

(iv) We include the rows that are marked by Δ and delete the rows that redundant.

3.2 (b) Formation of reduced matrix

Now we go to reduce the above modified cost matrix into an equivalent canonical matrix with elements.

$d'_{ij} = d_{ij} - \alpha_i - \beta_j$, Where $\alpha_I = {}^{min}_{j} d_{ij}$ and $\beta_j = {}^{min}_{i}(d_{ij} - \alpha_i)$

i.e. $\sum_{i \neq j} d_{ij} x_{ij} = \sum_{i \neq j} d'_{ij} x_{ij} +$ $[\sum \alpha_i + \sum \beta_j = \gamma,$ const.]

and so now we have a matrix having at least one zero in each row and column, γ is said to be bias of the matrix.

3.2 (c) Formation of alphabet table

Now we form the alphabet table writing S-D (Station – Distance) in columns and rows as stations. We write the travelling cost in ascending order before every station, in the manner Station - Distance.

3.2 (d) Lexicographic search

Step 0:- For the precedence constraints, say, $i_r < j_r$; r = 1,2,….s, put $d_{j_r i_r} = d_{i j_r} = d_{i_r j}$ As large as possible. Remove the 'bias'. This bias removal 'reduces' the cost matrix to a non-negative matrix with at least one, zero in each row and in each column (cf, Table-4.3). Obviously, it is enough to solve the problem with respect to this cost matrix. Sort each row in ascending order of distance and store the corresponding indices in another matrix, Alphabet Matrix, say X. Initialize the 'current trial solution value' to a large number. Since our starting node is 1, we start our computation from 1st row. Put s=1.

Step 1:- Go to the "s" element of the row (say, node p) and compute the cost of travelling. If the cost is greater than or equal to the 'current trial solution value', go to step 8, else, go to step 2.

Step 2:- If the (incomplete) word forms a sub-tour or if any prescribed restriction is violated, drop the city added in step 1 and increment " s" by 1, and then go to step 6; else, go to step 3.

Step 3:- If one full cycle is generated, then replace the cost of travelling as the 'current trial solution value' and go to step 8, else go to step 4.

Step 4:- Calculate the bound.

Step 5:- If the (Bound + Travel cost) is greater than or equal to the trial solution, drop the city added in step 1 and increment s by 1, and then go to step 6; else, go to step 7.

Step 6:- If k is less than (total number of nodes), go to step 1, else, go to step 8.

Step 7:- Go to sub-block, i.e., go to p^{th} row and then put s=1; go to step 1.

Step 8:- Jump this block, i.e., go to the previous row (i.e., node) and increment "s" by 1, where s was the column number of that row. This will automatically reject all the subsequent words from this block as solutions worse (at least, not better) than the current trial value. If the present node is the starting node and s = n, go to step 9, else, go to step1.

Step 9:- Current word gives the optimal tour sequence, with 'current trial solution value' as the optimum distance, with respect to the 'reduced' cost matrix.

Step 10:- Add the 'bias' to the optimal solution value obtained above and stop.

We again find another cycle and the feasibility of cycle, by finding TRVL and bound + Traveling cost < TRVL for each path.

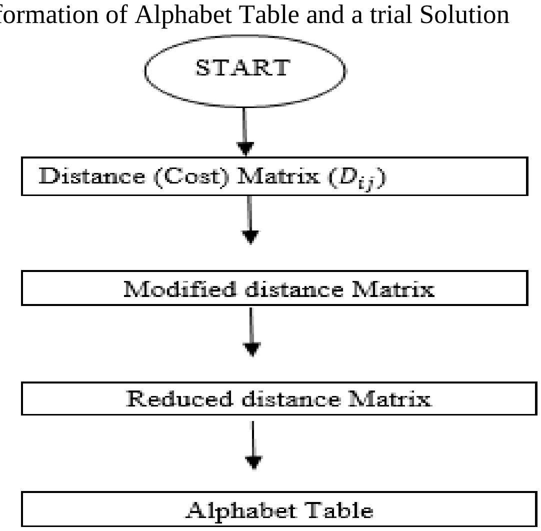
The symbols used therein are listed:

GS : Go to sub block i.e. attach the first 'free' letter to the current leader db' leads to 'dba'. JB: Jump the block 'abc' leads to 'abd' .

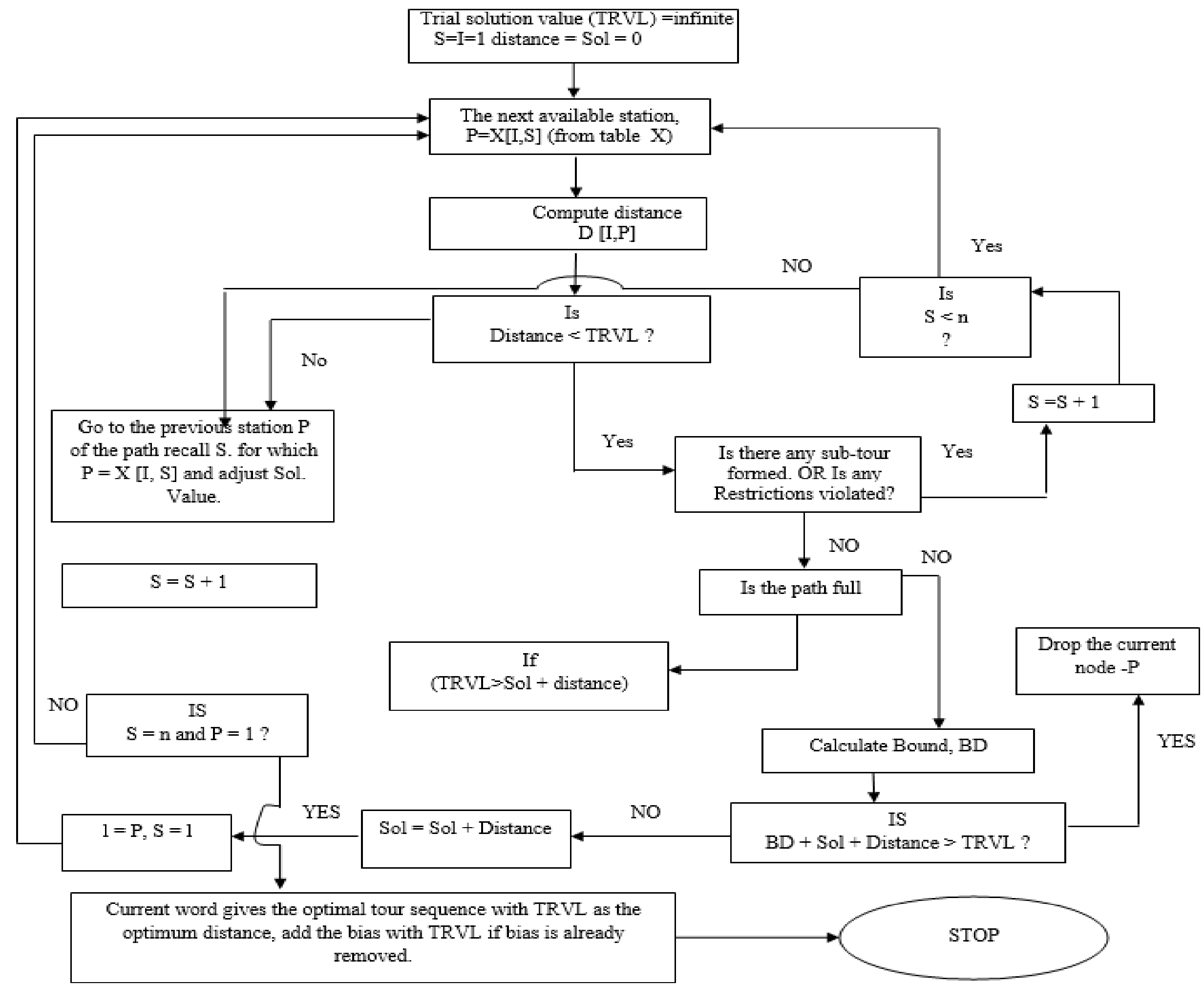
JO: Jump out to the next.

Bound :- Bound is the sum of the travel costs of the cities (not in the current word, excluding latest city) to the first reachable city (excluding the cities of current word, excluding home station.

3.2 (e) The flow chart for the algorithm .Part 1 formation of Alphabet Table and a trial Solution



Part 2: Lexicographic search



4. NUMERICAL ILLUSTRATION

We consider here a travelling salesman problem with multiple job facilities (J_1 J_2....J_m) available at various stations and a precedence constraint 4<6 is given. This is a six city problem. We desire to find it's optimal solution via lexicographic search. Whose distance and job matrix is given as following.

Table 4.1 : Distance and Job Matrix

Jobs	Station	1	2	3	4	5	6
	1	999	90	999	98	12	19
J_1, J_2	2	999	999	50	27	28	63
J_1, J_4,J_6	3	99	999	999	86	57	50
J_1, J_2, J_3	4	73	4	56	999	90	55
J_4, J_5	5	22	30	20	37	999	30
J_5, J_6,J_7	6	30	65	58	15	7	999

Table 4.2 : Check List

Jobs → Station ↓	J1	J2	J3	J4	J5	J6	J7
1	-	-	-	-	-	-	-
2	Δ	Δ					
3	Δ			Δ		Δ	
4	Δ	Δ	□				
5				Δ	Δ		
6					Δ	Δ	□

We can observe station 2 can be deleted 4,6 are necessary and 3 or 5 can be used Once we take 3 and delete 5 and 2

Table 4.3 : Modified Distance Matrix

Station	1	3	4	6	α_i
1	999	999	98	999	98
3	99	999	86	50	50
4	999	56	999	55	55
6	30	58	999	999	30

β_j	0	1	0	0	$\gamma = \sum \alpha_i + \sum \beta_j = 234$

Table4.4: Reduced Distance Matrix Station

Station	1	3	4	6	α_i
1	901	900	0	901	98
3	49	948	36	0	
4	944	0	944	0	
6	0	27	969	969	
β_j	0	1	0	0	$\gamma = \sum \alpha_i + \sum \beta_j = 234$

Table 4.5 : Alphabet Table : X

Station	S-D	S-D	S-D	S-D
1	4-0	3-900	6-901	1-901
3	6-0	4-36	1-49	3-948
4	3-0	6-0	1-944	4-944
6	1-0	3-27	4-969	6-969

S= Station number, D Distance

Table 4.6 : Search Table

$1 \to \alpha 1$	$\alpha 1 \to \alpha 2$	$\alpha 2 \to \alpha 3$	$\alpha 3 \to \alpha 4$	Additive Bound	Remarks
$1 \to 4_{(0)}$ (0)				0+0=0	GS
	$4 \to 3_{(0)}$ (0)			0+0=0	GS
		$3 \to 6_{(0)}$ (0)		0+0=0	GS
			$6 \to 1_{(0)}$ (0)	TRVL=0	JO
	$4 \to 6_{(0)}$ (0)			0+0=0	GS
		$6 \to 3_{(27}$ (27)		Distance>TRVL	JO
$1 \to 3_{(900)}$ (900)				Distance>TRVL	STOP

From the above table we can observe that one feasible solution is

1→ 4 → 3 → 6 →1→ Feasible Solution No. 1

and the total travelling cost is 234.

As we can observed in Table 4.2 check list that if we choose stations 5 in place of station 3, in Modified distance matrix Table 3.3 all jobs are completed. So, we may replace the station 3 by Station 5.

Table – 4.7 Modified Distance Matrix.

Station	1	4	5	6	α_i
1	999	98	12	999	12
3	999	999	90	55	55
4	22	37	999	30	22
6	30	999	7	999	7
β_j	0	15	0	0	$\gamma = \sum \alpha_i + \sum \beta_j = 111$

Table - 4.8 Reduced Distance Matrix

Station	1	4	5	6	α_i
1	987	71	0	987	12
4	944	929	35	0	55
5	0	0	977	8	22
6	23	977	0	992	7
β_j	0	15	0	0	$\gamma = \sum \alpha_i + \sum \beta_j = 111$

Table 4.9 Alphabet Table : X

Station	S-D	S-D	S-D	S-D
1	5-0	4-71	6-987	1-987
4	6-0	5-35	4-929	1-944
5	1-0	4-0	6-8	5-977
6	5-0	1-23	4-977	6-992

S= Station number, D = Distance

Table - 4.10 Search Table

$1 \to \alpha 1$	$\alpha 1 \to \alpha 2$	$\alpha 2 \to \alpha 3$	$\alpha 3 \to \alpha 4$	Additive Bound	Remarks
$1 \to 5_{(0)}$ (0)				0+23=23	GS
	$5 \to 4_{(0)}$ (0)			0+23=23	GS
		$4 \to 6_{(0)}$ (0)		0+23=23	GS
			$6 \to 1_{(0)}$ (0)	TRVL=23	JO
$1 \to 4_{(71}$ (71)				Distance>TRVL	STOP

So the other feasible cycle is 1→5→4→6→1

Total travelling cost of above cycle 23+111 = 134 →Feasible Solution 2

So the min cost to solution 1st and Solution 2nd is solution 2nd as the minimum cost in 2nd solution.

So the optimal path to the given problem is 1→5 →4→6→1 with travelling cost as 134.

5.CONCLUSION

The procedure developed by others Algorithms available in literature requires more variable and more constraint, to solve time dependent traveling salesman problem, than their requirement for author's procedure. The procedure developed by author takes care of the simple combinatorial structure of traveling salesman problem. So have lesser calculations. A computer program may be developed according to algorithm.

REFERENCES

[1] Affenzeller, M., Wanger, S., "A self-adaptive model for selective pressure handling within the theory of genetic algorithms", EUROCAST 2003, Las Palmas de Gran Canaria, Spain, Lect. Notes Comp. Sci. 2809 (1), 384–393. (2003)

[2] Bianco L., Mingozzi S." Exact and hehristic procedures for the TSP with precedence constraints, based on dynamic programming "Information Systems &Operational Research,32,19-32,(1994).

[3] Bianchi, L., Knowles, J., Bowler, J.," Local search for the probabilistic traveling salesman problem: Correction to the 2- p-opt and 1-shift algorithms", Eur. J. of Oper. Res. 162(1), 206–219. (2005)

[4] Carpaneto, G., Toth, P.," Some new branching and bounding criteria for the asymmetric traveling salesman problem", Management Science 21, 736–743. (1980)

[5] Chu, S.C., Roddick, J.F., Pan, J.S.," Ant colony system with communication strategies", Inform. Sci. 167 (1–4), 63– 76. (2004)

[6] G. K. Jati and S. Suyanto, "Evolutionary discrete firefly algorithm for travelling salesman problem," in Adaptive and Intel-ligent Systems, vol. 6943 of Lecture Notes in Computer Science, pp. 393–403, Springer, (2011).

[7] Jun-man, K. and Yi, Z. "Application of an improved ant colony optimization on generalized traveling salesman problem". Energy Procedia, 17, Part A(0), 319 – 325.(2012)

[8] Liu, G., He, Y., Fang, Y., Oiu, Y.," A novel adaptive search strategy of intensification and diversification in tabu search", in: Proceedings of Neural Networks and Signal Processing, Nanjing, China.(2003).

[9] Onwubolu, G.C. and Clerc, M.," Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization", Int.J. Prod. Res. 42 (3), 473–491. (2004)

[10] Sobhan Babu,K & Sundara Murthy.M, "An efficient algorithm for Variant Bulk Transportation Problem", International Journal of Engineering Science and Technology, Vol.2(7), pp.2595-2600.(2010)

[11] Sobhan Babu,K ,Keshava redid.E & Sundara Murthy.M" an exact algorithm for travelling salesman problem" Vol.3(1).pp.317-321.(2012)

[12] Thanaboon Saradatta, Pisut Pongchairerks A "Time-Dependent ATSP With Time Window and Precedence Constraints in Air Travel" Journal of Telecommunication, Electronic and Computer EngineeringVol.9, No.2-3 pp 149-153. (2017).

[13] Turkensteen, M., Ghosh, D., Goldengorin, B., Sierksma, G., "Tolerance-based Branch and Bound algorithms for the ATSP", European Journal of Operational Research 189: 775–788, (2007) Available online at www.sciencedirect .com.