ISSN 0976 – 1411

Available online at www.internationaleJournals.com

International eJournals

INTERNATIONAL eJOURNAL OF MATHEMATICS AND ENGINEERING

www.InternationaleJournals.com

International eJournal of Mathematics and Engineering 6 (2016) Vol. 7, Issue 2, pp 42 – 50

A Lexisearch Approach to the Travelling Salesman Problem

[1]Ch.N. Anuradha, [2]V.V.Haragopal and [3]Shahnaz bathul

[1]Department of Mathematics, Vasavi College of Engg., Hyderabad, Telangana, India. E-mail: chnanuradha@yahoo.com

[2]Department of Statistics and Center For Quantitative Methods, Osmania University, Hyderabad, Telangana, India. E-mail: haragopal_vajjha@yahoo.com

[3]Department of Mathematics, JNTU, Hyderabad, Telangana, India. E-mail: sbathul@gmail.com

Abstract: In this paper we introduce a simple Lexi-search Algorithm for a Travelling Salesman problem. The Lexi-search derives its name from lexicography.This approach has been used to solve various combinatorial problems efficiently. This algorithm is deterministic and is always guaranteed to find an optimal solution.

Keywords: Introduction, Combinatorial problem,Lexi-search Algorithm to travelling Salesman problem,conclusion.

1. INTRODUCTION :

Mathematical programming is concerned with finding optimal solutions ,via lexisearch algorithm. obtaining good solutions. The set of all possible solutions to a problem is arranged in hierarchy like words in a dictionary, such that each incomplete word represents the block of words with this incomplete word as the leader of the block. Bounds are computed for values of the given function over these blocks of words. These are compared with "best value" found so far. If no word in the block can be greater than the "best value", jump over the block to the next one. However, if the bound indicates a possibility of better solution in the block, enter into the minor block by concatenating the present leader with appropriate letter and set a bound for the minor block so obtained.

2. COMBINATORIAL PROBLEM:

If the solution space of a problem consists of combinatorial data ,then it is known as combinatorial problem.

An important class of Discrete Programming Problems is 'Combinatorial programming problems' (Sundara Murthy - 1979). A problem of this class is associated in selection of a solution which optimizes an objective function over the set of alternatives that satisfy the feasibility criterion i.e. over the feasibility region.

The Travelling Salesman Problem.

A salesman wishes to visit cities A,B,C,D & E. He does not want to visit any city twice before completing his tour of all the cities and wishes to return to the point of starting journey.To solve the combinatorial problems different techniques are available. However no general approach which is suitable for all discrete programming problems, seems to be available and the methods of finding optimal solutions mainly by search methods (Pandit 1963). For many of the combinatorial programming problems the solution space is finite and hence, it is theoretically examined all the solutions.

a) Breach and bound algorithms (little et al, 1963)
b) Lexicographic search (Pandit 1963)

II. Lexisearch Method

The Lexisearch approach was first proposed by Pandit (1962) in the contest of 'The Loading Problem'. From 1963 onwards this approach has been used to solve various combinational problems efficiently, e.g., the Assignment problem (jain, Mishra and Pandit, 1964). The staff location problem (Das 1976), the Travelling Salesman Problem (Pandit and Rajbongshi, 1976, Sundara Murthy, 1979; Subramanyam, 1980, Ramesh 1980, Chandrasekhar Reddy, 1987). The job scheduling problem (gupta, 1967; Rajbongshi, 1982; Bhanumurthy, 1986) In all these problems the lexicographic search was found to be more efficient than the Branch bound algorithms (c.f.Ravi Kumar, 1989). It is viewed that in general sense, the branch bound approach can be viewed as a particular case of lexicographic search.

A systematic version of this Branch and Bound was developed and this is known as **lexicographic Search Technique** by Pandit - 1962 & 1963. It is also worth noting that Branch and Bound technique can be viewed as a particular case of Lexicographic Search Technique. A detailed discussion on this can be found in the sequel.

Travelling salesman problem (TSP, for short) is one of the oldest combinatorial programming problems as defined earlier; (Flood, 1956 and Croes, 1958) can be stated as follows (e.f.sundara Murthy, 1979). The objective of TSP is to find a minimal length tour i.e., a tour, which is of minimal length. K The TSP can be defined equivalently as a problem of finding Hamiltonian cycle of the shortest length in a connected weighted graph. M .Neril Flood who was associated with RAND Corporation gave wide publicity to the TSCP. TSP has some similarity with those other problems but is much harder to solve.

The cutting plane method was used to solve the above LPP with integer constraints and they also seem to have used the concept of branch and bound in some sense to solve the same (e.f.Lawler et al 1985)

There are certain approaches for exact solution of TSP and they are

1. Integer programming approach
2. Dynamic programming approach
3. Branch and bound algorithm
4. East man's algorithm
5. Lexisearch Method.

The Lexisearch Approach

The lexicographic search approach to the combinatorial optimization problem, as already pointed out was developed by Pandit in the first instance for the Knapsak problem (Pandit, 1962) and has since been applied to many other combinatorial problems like job scheduling (ef.Gupta, 1969) etc., in addition to the assignment problem and travelling salesman problem (ef. Das. 1976).

Go into the 'sub blocks' if the block-bound is less than the trial solution value and hence, the current block may contain a solution better than the trial solution value.

1. Jump over to the 'next' Block'; if he block-bound is greater than the trial solution value, or
2. Jump out to the next 'super block' if the current block, which is to be jumped over is the last block of the present super block.

Further, if the value of the current leader is already greater or equal to the current trial value, no need for checking the subsequent blocks with in this super-block. These concepts are illustrated below: in case of the TSP.

Let a, b, c, d,e be the four cities to be travelled by a salesman, then the set of possible partial and complete word is listed lexicographically as follows:

GS - Go to Sub-block.

JB - Jump the current block.

JO - Jump out Next block.

BS - Best Solution.

TV – Total Value.

B - Bounds.

Algorithm LEXIG TSP

Step 1: Arrange all cities according to their distances from the city i=1, 2… N+ (k-1). This arrangement consists of N+(k-1)columns and N-1 rows since the origins are removed. K Each column represents a city 'A' and the elements in that column are the cities arranged in increasing order according to their distance from the City 'A'.

Step 2: Include the first reachable city from the origin in the partial solution. 'W'. If the distance itself is greater than or equal to TR then stop. Otherwise go to next step.

Step 3: calculate the Bound.

Step 4: If the (Bound+ Partial solution value) is greater than or equal to trial solution then drop the city added in step Now go to step 2 i.e., JB otherwise go to next step i.e., GS.

Step 5: Include the next reachable city (from the last city included in the partial solution 'W') into the partial solution.

Step 6: If partial solution value is greater than or equal to the TR, drop the city added in step 5, also drop the last city in 'W' go to step 5 (i.e., JO) otherwise goes to step 7.

Step 7: If the (Partial solution value + Bound) is greater than or equal to TR, then drop the newly added city in step 5 (i.e., JB) Go to step 5. Otherwise go to step 8.

Step 8: If the partial solution contains…….. cities, add the dummy origin to the partial solution. Calculate the value of this partial solution. If it is greater than or equal to the TR, drop the dummy origin and last two cities from the partial solution(i.e., JO). If 'W' contains only one city, go to step 2; otherwise go to step 5, Otherwise go to next step.

Step 8: Calculate the Bound.

Step 10: If the (Partial solution value+bound) is greater than or equal to TR then drop the dummy origin and also last city from 'W' go to step 5(i.e., JB) Otherwise go to step 11.

Step 11: Include the latest possible city from the dummy origin I in 'W'

Step 12: If (partial solution value) is greater than or equal to TR, drop te last dummy origin and also the city from which the ith dummy origin was reached. K Go to step 5. Otherwise go to next step.

Step 13: Calculate the Bound.

Step 14: If (Partial Solution value + Bound) is greater than or equal to TR, dropl the recently added city in 'W' and go to step 11. Otherwise go to next step.

Step 15: Include the latest reachable city from the last city W.

Step 16: If (partial solution value) is greater than or equal to TR, drop the last two cities; If dummy origin happens to be one of these two cities, also drop the city from which the dummy origin was reached, reduce the value off I by I i.e., I becomes i-1 otherwise go to next step.

Step 17: Calculate the Bound.

Step 18: If (Partial Solution value + Bound) is greater than or equal to TR, from the latest city i.e., JB. Go to step 15. Otherwise go to next step.

Step 19: If the number or elements in W is less than …… go to step 15. Otherwise go to step 20.

Step 20: Add the dummy origin to W and calculate the value of W. If it is greater than or equal to the TR drop the dummy origin and also the last two cities in W. Go to step 15. Otherwise go to step 21.

Step 21: Replace TR by Partial Solution value and trial solution by W. Drop the dummy origin and last two cities in W. Go to step 15.

The symbols used in table III are listed below:

GS: go to sub-block i.e., attach the first 'free' letter to the current leader.

GS for 'db' leads to 'dba' as augmented leader.

JB or GNSB: Jump the block i.e., go to the next block of the same order i.e., Replace the last letter of the current block by the letter next to it in the alplhaber table. JB or GNSB (GO TO NEXT SUPER BLCOK) FOR 'acb' 'adb'

JO: jump out to the next, higher order block i.e., drop out the last letter of the current letter and then jump the block.

JO for 'cdbe' is 'cde'

TRVL: currently trial solution value.

CR: Column repletion.

Vt: Trial Value.

Table I:

Consider a 5x5 city problem.

	1	**2**	**3**	**4**	**5**
1	∞	**2**	**5**	**7**	**1**
2	**6**	∞	**3**	**8**	**2**
3	**8**	**7**	∞	**4**	**7**
4	**12**	**4**	**6**	∞	**5**
5	**1**	**3**	**2**	**8**	∞

Table II : Alphabet Table

	1	**2**	**3**	**4**	**5**
1	**5(0)**	**2(1)**	**3(3)**	**4(6)**	**1(∞)**
2	**3(0)**	**5(0)**	**1(4)**	**4(6)**	**2(∞)**
3	**4(0)**	**2(3)**	**5(3)**	**1(4)**	**3(∞)**
4	**2(0)**	**3(1)**	**5(1)**	**1(8)**	**4(∞)**
5	**1(0)**	**3(0)**	**2(2)**	**4(7)**	**5(∞)**

Table III: Lexisearch Table:

					Sol	Bound	TV	BS	Remarks
GS	1--5				0	0	0	∞	Ok
GS		5--3			0	4	4	∞	Ok
GS			3--4		0	4	4	∞	Ok
GS				4--2	0	4	4	∞	Ok
JO					4	0	4	∞	ok

……

					Sol	Bound	TV	BS	Remarks
GS	1—2				1	1	2	4	Ok
GS		2--3			1	1	2	4	Ok
GS			3--4		1	1	2	4	Ok

GS			4—5			2	0	2	4	Ok
JB				5--1		2	0	2	4	ok

So consider 4 as a BS

……

					Sol	Bound	TV	BS	Remarks
JB	1—3				3	--	--	2	Change
JO	1---4				6	--	--	2	Change

So consider 2 as a BS

					Sol	Bound	TV	BS	Remarks
GS	2—3				0	1	1	2	OK
GS		3—4			0	1	1	2	OK
JB			4--5		1	1	2	2	change
JB			4--1		9	-	-	2	change
JB		3--5			3	-	-	2	change
JB		3--1			4	-	-	2	change
JB	2--5				0	1	1	2	OK
GS		5—1			0	3	3	2	change
JB		5—3			0	10	10	2	change
JB		5—4			7	-	-	2	change
JB	2—1				4	-	-	2	change
JO	2—4				6	-	-	2	change
GS	3—4				0	0	0	2	OK
GS		4—2			0	0	0	2	OK
JB			2—5		0	3	3	2	change
JB			2--1		4	-	-	2	change
JB		4—5			1	1	2	2	change
JB		4—1			8	-	-	2	change

JB	3—2					3	-	-	2	change
JB	3—5					3	-	-	2	change
JO	3--1					4	-	-	2	change
						Sol	Bound	TV	BS	Remarks
GS	4—2					0	0	0	2	OK
GS		2—3				0	0	0	2	OK
JB			3--5			3	-	-	2	change
JB			3--1			4	-	-	2	change
JB		2--5				0	3	3	2	change
JB		2--1				4	-	-	2	change
JB	4--3					1	3	4	2	change
JB	4--5					1	1	2	2	change
JO	4--1					8	-	-	2	change

……..

						Sol	Bound	TV	BS	Remarks
GS	5—1					0	1	1	2	OK
JB		1—2				1	1	2	2	change
JB		1--3				3	-	-	2	Change
JB		1--4				6	-	-	2	change
GS	5--3					0	0	0	2	OK
GS		3—4				0	0	0	2	OK
GS			4--2			0	0	0	2	OK
JB				2—1		4	-	-	2	change
JB			4—1			8	-	-	2	change
JB		3—2				3	-	-	2	change
JB		3—1				4	-	-	2	change
JB	5—2					2	-	-	2	change
JO	5—4					7	-	-	2	change

For, the above search table the optimal solution is 15, and then the complete word which belongs to the sequence is 1→2→3→4→5→ 1

III. Conclusion

A Lexisearch allows parallelization of the algorithm as compared to the Branch and bound algorithm. In terms of the complexity also Lexisearch appears quite competitive as reported by many investigators on the basis of simulation studies. Recently, the approach of Lexisearch methodology is established in various fields of operations research and this method is being tried in parallel computing. This algorithm is deterministic and is always guaranteed to find an optimal solution, unlike the conventional dynamic programming or branch and bound algorithms, which require exponential space; the lexicographic algorithm required only a linear space with respect to the problem size. This algorithm is easily parallelizable and found that this method is superior to the existing methods.

References:

[1]. M. Ramesh. *"A lexisearch approach to some combinatorial programming problems"*, University of Hyderabad, India, 1997.

[2]. S.N.N. Pandit. "*Some quantitative combinatorial search problems*", Indian Institute of Technology, Kharagpur, India, 1963.

[3]. Srinivasan V. & G.L. Thompson (1973). An Algorithm for Assigning Uses to Sources in a Special Class of Transportation Problems, Op. Res., Vol. 21, No.1.

[4]. Subramanyam, Y. (1980): Scheduling, Transportation & Allied Combinatorial Programming Problems, Ph.D. thesis, REC Warangal, India (Unpublished).

[5]. Kanthi Swarup ,P.K.Guptha ,Man Mohan-"*Operations Research*"- Sulthan Chand and Sons 13th Edition.

[6]. Hamdy A.Taha-"*Operations Research-An Introduction*"-(7th edition),Pearson Education (Singapore).

[7]. J.K .Sharma -"*Operations Research-Theory and applications*"-MacMillan India Ltd.

[8]. S.C. Sharma –"*Introductory operations Research*"- e books.

[9]. P.Rama Murthy-" *Operations Research linear programming"*-e books.

%%%