ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Review of Travelling Salesmen Problem through Lexi Search Method

Nabeel Naeem Hasan
Masters in Mathematics, Student
Department of Mathematics,
College of Science,
Osmania University, Hyderabad, India.

ABSTRACT

The aim of this paper is to introduce Lexisearch the structure of the search algorithm does not require huge dynamic memory during execution. Mathematical programming is concerned with finding optimal solutions rather than obtaining good solutions. The Lexisearch derives its name from lexicography .This approach has been used to solve various combinatorial problems efficiently, the Assignment problem, the Travelling Salesman Problem, the job scheduling problem etc.

In all these problems the lexicographic search was found to be more efficient than the Branch bound algorithms. This algorithm is deterministic and is always guaranteed to find an optimal solution. The problem of mathematical programming is to find the maximum or minimum of an objective function whose variables are required to satisfy a set of well-defined constraints. If the objective function is continuous in the variable values that also lie in a feasible region, which is compact, then it is a continuous programming problem. Where as if the decision variables take only discrete values then it is known as discrete programming problem. If the set of solutions is a finite discrete set or not necessarily of variables in the usual sense but, May even be of other entities like permutations or combinations, then the problem is one of discrete programming. Operations research problems are outlined slightly combinatorial and non-combinatorial having a common feature in that the objective function is to be minimized is the maximum of a set of function values.

1.1 INTRODUCTION:

A mathematical programming problem solving is concerned with discovering ideal arrangements as opposed to acquiring great arrangements. The idea of improvement is antiquated and has quickened hugely with the advancement of machines and direct programming in the late 1940‟s. The numerical programming issues are comprehensively grouped into two classes specifically:

1. Continuous programming issues

2. Discrete programming issues.

The issue of mathematical programming is to discover the greatest or least of a target work whose variables are obliged to fulfill a set of decently characterized obligations.

In the event that the target capacity is constant in the variable values that likewise lie in a plausible area, which is minimal, then it is a consistent programming issue. This kind of issues are fathomed by simplex methodology (Hadely 1994), Lagrangian multipliers system (Rao, 1984), where as though the choice

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

variables take just discrete values then it is known as discrete programming issue. In the event that the set of arrangements is a limited to discrete situated or not so much of variables in the standard sense yet, may even be of different substances like stages or mixes, then the issue is one of discrete programming. Operations research issues are delineated somewhat combinatorial and non-combinatorial having a typical peculiarity in that the target capacity is to be minimized is the greatest of a set of capacity qualities.

Combinatorial issue: If the arrangement space of an issue comprises of combinatorial elements, then it is known as combinatorial issue. As indicated by Pandit (1963), the combinatorial programming issues can be portrayed as: "There is a numerical capacity characterized over the area of courses of action or choices of a set of components. There are additionally possibility criteria .Now, the issues to discover the courses of action which are plausible and which enhance the numerical capacity".

The combinatorial issues are not new to arithmetic; maybe one of the methods for describing them is by saying that they are concerned examples, normally arrangements that can be discovered of a limited number of components. the regular combinatorial issues are concerned with identification of examples fitting in with a given structure; class-significant zones of diagram hypothesis are of this type. Like the numerical issues by and large, in combinatorial programming additionally we are less keen on the quantity of examples of a given sort, yet in partner in any case, a numerical quality is to each example of investment and afterward asking regarding how to get that example which minimizes (or maximizes)this numerical worth, among a decently characterized set(essentially of a limited cardinality) of examples.

As it were, "the scope of the variable of the combinatorial capacities" is the set of examples frequently changes, mixes are more convoluted image chains and the exceptionally advantageous ideas of neighboring smoothness and related thoughts are totally immaterial in this minimax improvement issue. A few sorts of combinatorial programming issues have the markovian emphasize as in they can be consecutively handled, with the structure empowering that the area of arrangement resulting to one phase of the arrangement is autonomous of the prior piece of arrangement yet depends just on the current stage arrangement. It can be related to cooperatively be the phases of arrangements and when arrangements of the issue can be figured interms of a appropriate operation, it will be seen that the concerned operation is affiliated and is of extraordinary hugeness numerically. If not additionally hypothetically, settling the issue. Numerous combinatorial programming issues shockingly don't fall in this classification they are non-markovian in that at any phase of arrangement, lingering piece of the issue can't be connected from the halfway arrangement up to that stage. A traditional illustration of this sort is the Travelling salesperson and the Assignment issues. One has maybe, to be content with an anticipated calculation to tackle with the issue however as in Assignment issue, the combinatorial structure may prompt fascinating side results, which, if one is fortunate, can likewise be of some combinatorial noteworthy.

Truth be told, the range of combinatorial writing computer programs is loaded with such issues with some defense, we might hypothetically say that the bringing together element which ties all these issues into one class is that they can't be characteristically brought under one head by any positive definition, and maybe the main method for handling is by contriving suitable calculations focused around thoughts of hieratically organized numerical set of capacities i.e., designs, that can't be orchestrated in an order in the estimation of the capacity which prompts the Lexisearch and extension bound calculations. Correlation of calculations for arrangements of any such issues is fairly troublesome and is frequently certain judgments about relative weight age to be given to distinctive segments of calculation. It is standard to disregard the quantity of augmentations and subtractions however divisions in the execution of

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

the calculation. Combinatorial programming calculations frequently include a lot of augmentation on the setting of spotting in the machine memory the different dimensional variables. In the previous one may be compelled to acknowledge the certain yet exceptionally unacceptable feedback of the time needed for taking care of various issues with haphazardly produced information on any accessible machine, which was moderate. Anyhow the present situation is truly diverse where we can create various issues of different sizes and explain then with no time, due to the quick processing office. An issue, which is basically not combinatorial, is the one where calculations for its answers are produced in the investigative way utilizing coherence ideas, yet physically it creates the impression that there ought to be an ideal answer for any self-assertive issue of this sort.

Nonetheless, it is not generally conceivable to perceive a proposed arrangements, one can characterize a set of homogeneous mathematical statements with a few variables limited to be negative and ask whether the framework has arrangement. In the event that the answer is YES, the arrangement can be fundamentally being made strides. On the off chance that the answer is NO it is an ideal arrangement. In this way, different creators addressed this question one way or other by a couple of trials changing the arrangement without altering the minimax worth and eventually having the capacity to get an identifiable ideal arrangement.

1.2 TRAVELLING SALESMAN PROBLEM:

The origins of the travelling salesman problem are unclear. A handbook for travelling salesmen from 1832 mentions the problem and includes example tours through Germany and Switzerland, but contains no mathematical treatment.

Mathematical problems related to the Travelling salesman problem were treated in the 1800s by the Irish mathematician W. R. Hamilton and by the British mathematician Thomas Kirkman. Hamilton's Icosian Game was a recreational puzzle based on finding a Hamiltonian cycle. The general form of the TSP appears to have been first studied by mathematicians during the 1930s in Vienna and at Harvard, notably by Karl Menger, who defines the problem, considers the obvious brute-force algorithm, and observes the non-optimality of the nearest neighbor heuristic.In the 1950s and 1960s, the problem became increasingly popular in scientific circles in Europe and the USA.

Notable contributions were made by George Dantzig, Delbert Ray Fulkerson and Selmer M. Johnson at the RAND Corporation in Santa Monica, who expressed the problem as an integer linear program and developed the cutting planemethod for its solution. With these new methods they solved an instance with 49 cities to optimality by constructing a tour and proving that no other tour could be shorter. In the following decades, the problem was studied by many researchers from mathematics, computer science, chemistry, physics, and other sciences.

Richard M. Karp showed in 1972 that the Hamiltonian cycle problem was NP-complete, which implies the NP-hardness of TSP. This supplied a scientific explanation for the apparent computational difficulty of finding optimal tours.Great progress was made in the late 1970s and 1980, when Grötschel, Padberg, Rinaldi and other managed to exactly solve instances with up to 2392 cities, using cutting planes and branch-and-bound.In the 1990s, Applegate, Bixby, Chvátal, and Cook developed the program Concorde that has been used in many recent record solutions.

Gerhard Reinelt published the TSPLIB in 1991, a collection of benchmark instances of varying difficulty, which has been used by many research groups for comparing results. In 2005, Cook and others computed an optimal tour through a 33,810-city instance given by a microchip layout problem, currently the largest solved TSPLIB instance. For many other instances with millions of cities, solutions can be found that are provably within 1% of optimal tour.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

1.3 LEXI SEARCH METHOD:

Lexicographic Search Approach is a systematized Branch and Bound approach, developed by Pandit in the context of solving a loading problem in1962i.e. Even before the Little et al came out with their Branch and Bound for the Travelling Salesman problem. This approach has been found to be fruitful in many of the Combinatorial Programming Problems.

In principle, it is essentially similar to the Branch and Bound method as adopted by Little et al -1963 and it is worth 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 a set of symbols A= (1, 2, 3,.....,n) and the different possible sequences of length k of these symbols. Thus (α1, α2, ...,αk) is 13a k-word formed from the alphabet of n symbols 1, 2, 3 ..... ,n, the i^{th} letter in the word is αiε A.

For a particular problem, we can define all solutions of the problem as a subset of possible words with this alphabet and attach a value to each of the words. By defining analphabetic 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 <= m are called

Incomplete words standing for the set or block of the (m-k)! Words of length k having the incomplete word (α1, α2...αk) as their leader. Numerical values can be associated for these words which have the property that as the word-length is increased by a concatenation i.e. attaching of more letters to the right of the word will monotonically non-decrease the values of words and an effective bound can be computed with relative case for the values of words belonging to a block defined by a leader. We can even generalize this, by considering m alphabets, A1, A2, A3...A m and defining the words (α1, α2,....., αm) where αiεAi, i = 1, 2, ..., m. After this, searching for an optimum word is 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.

This concept can be better understood by considering the Travelling Salesman problem. Travelling Salesman problem is one of the oldest of the combinatorial problems and its structure is close to that of the Assignment Problem with a difference that the solution set of the Travelling Salesman problem is rather more restrictive. That is, the permutation (solution) matrix is a feasible solution to the Assignment Problem if the matrix is non-decomposable then it is a solution to TSP.

2. TRAVELLING SALESMAN PROBLEM IN DIAGRAMMATIC APPROACH:

A breakthrough came when George Dantzig, Ray Fulkerson, and Selmer Johnson (1954) published a description of a method for solving the TSP and illustrated the power of this method by solving an instance with 49 cities, an impressive size at that time. They created this instance by picking one city from each of the 48 states in the U.S.A. (Alaska and Hawaii became states only in 1959) and adding Washington, D.C.; the costs of travel between these cities were defined by road distances.

Rather than solving this problem, they solved the 42-city problem obtained by removing Baltimore, Wilmington, Philadelphia, Newark, New York,

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Hartford, and Providence. As it turned out, an optimal tour through the 42 cities used the edge joining Washington, D.C. to Boston; since the shortest route between these two cities passes through the seven removed cities, this solution of the 42-city problem yields a solution of the 49-city problem.

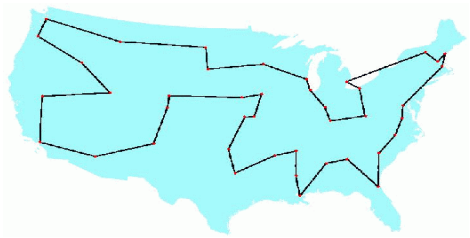


Figure 1.1: 48states of USA

3.1 LEXISEARCH APPROACH:

The travelling salesman problem with the following restrictions was solved by Scroggs and Tharp (1972) and Das Shila(l976).

i. Some ordered pairs of cities are given such that the salesman should visit the first city before the second; this is called the 'precedence' constraint
ii. Some cities are specified as to be visited at specific steps from the headquarters; this is called the 'fixed position' constraint.

In the present chapter we will study the following generalisation which can be called the 'Truncated Travelling Salesman Problem.

There are n cities and N={1,2,3,….,n} The distance d(i, j) between any pair of cities (i,j) is known. A subset of the n cities HQ constitutes the potential places for setting up a headquarters. A salesman has to visit only m out of the n cities, with the restriction that his tour should include at least one city from HQ. The problem is to find a feasible tour of m cities with a minimum length.

At the outset the above problem can be thought of as choosing all possible sets of m cities from N which includes a city from HQ and solve this as a m-city, travelling salesman problem. In this case the number of problems will be

n C_m-(n-h) C_m where h=[HQ]

Obviously the number grows very high for even moderate values of m and n. Hence solving the above problems as a series of m-city salesman problems win be impracticable. In the sequel, we will develop a lexi-search algorithm, based on 'Pattern Recognition Technique', to solve this problem.

The concepts and the algorithm developed will be illustrated by a numerical example for which n=8, m=5, and h=3. Let N={1,2,3,4,5,6,7,8} and HQ = {1,4,7}.

The distance matrix D is given as Table 3.1.

Table 3.1

∞	89	40	13	37	38	74	13
47	∞	07	13	52	89	63	76
12	62	∞	23	40	57	17	18
74	05	11	∞	28	48	84	12
34	12	46	98	∞	37	15	41
37	20	42	68	55	∞	41	33
35	23	96	27	89	23	∞	08
01	30	41	80	30	77	38	∞

d (i,j), i = 1,2,.,…,8, are taken as ∞, as they are irrelevant in calculating a tour for the salesman. Though d(i, j)e are taken as positive integers, we could have as well chosen any real value.

An n x n indicator matrix X = [x (i, j)= 0 or l] represents a trip-schedule for the salesman, in which x(i, j)=1 indicates that the salesman visits city J from city i, and if there is no such trip, it is indicated by x (i, j) = 0. X is called a 'solution'.

The indicator matrix X given by Table 3.2 is a solution to the numerical example, and represents the following trip-schedule.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Table 3.2

0	0	1	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0
1	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

The salesman visits cities 3 and 6 from city I, visits city 4. from city 3, visits city 6 from city 4 and visits cities 1, 7 and 8 from city 6. Obviously, this solution is not a feasible solution since the cities {1,3,4,5,6,7,8} which are seven in number are involved in this trip-schedule, whereas he has to visit only m=5 cities, and also there is no tour connecting all these cities. The matrix X given by the Table 3,3 is also not a feasible solution since there is no tour connecting the five cities {2,4,5,7,8}, which are in the trip-schedule defined by it. The matrix given in Table 3.4 is a feasible solution, which involves the five cities { 2,4,5,7,8} in the trip-schedule; there is a tour connecting these cities (8-5-7-2-4-8) and cities 4 and HQ.

Table 3.3

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	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	1
0	0	0	1	0	0	0	0

Table 3.4

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	1	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

3.2 DEFINITION OF PATTERNS:

An indicator matrix, which is associated with a trip-schedule, is called a 'pattern'. A pattern is said to be feasible if the matrix X is a feasible solution. The patte given by the tables 3.2 and 3.3 are not feasible whereas the pattern given by Table 3.4 is a feasible pattern. V(X), the value of the pattern X, is defined as

$$\mathbf{V(X)} = \sum_{i=1}^{m}\sum_{j=1}^{m} \boldsymbol{d(i,j),x(i,j)}$$

The words "pattern", "'pattern X", matrix X' and l word(which is defined later are used, in the sequel, synonymously V(X), for the pattern given by Table 3.2 is

$$V(X) = 40+38+ 23+48+34+16+41 = 239$$

V(X) for the patterns given by Tables 3.3 and 3.4 are respectively 147 and 93.

Each pattern X can also be represented by the set of all ordered pairs {(I, j)} for which x (i,j)=1 with the understanding that the value of the ordered (other) pairs is zero (vide 2.2).

Thus the ordered pairs set $\{(x_1, y_1)\}$, i=1,2,……,7

= {((l, 3), (1,6), (3,4), (4,6), (5,1), (5,7), (5,8)} represents the pattern X given by Table 3.2. Similarly, the sets of ordered pairs {(2,4), (4,2), (5,8), (7,8), (8,4)} and {(2,4), (4,8), (5,7), (7,2), (8,5)}

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

represent respectively, the patterns given by Tables 3.3 and 3.4.

There are n ordered pairs in a matrix X. For convenience there are arranged in an increasing order of their corresponding distances and are indexed from 1to n^2 (vide 2.2). Let B=(l,2,...,n^2) be the set of n^2 indices. Let BD be the corresponding set of distances. If α, β€B and α<β, than BD(α) ≤ BD(β).

Also lit the arrays R and C be the arrays of raw and column Indices of the ordered pairs represented by B and m be the array of the cumulative sums of the elements of BD. The value of the arrays B, BD, DD, R and C for the example is given as Table 3.5.

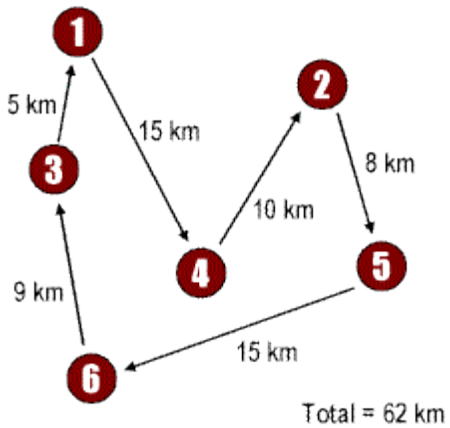
To illustrate the entries in the Table, consider 12GB. It represents the ordered pair

(R(12), C(12)) = (5, 7). Then BD (12) = d (5,7) = 15
and DD (12) = 122

4. APPLICATIONS OF TSP

In this chapter we list some of the TSP problems in the literature. Application TSP and its solution using different methods and techniques are illustrated here in this chapter.

Problem: Given a complete undirected graph $G=(V, E)$ that has non-negative integer cost $c(u, v)$ associated with each edge (u, v) in E, the problem is to find a hamiltonian cycle (tour) of G with minimum cost.



A salespersons starts from the city 1 and has to visit six cities (1 through 6) and must come back to the starting city i.e., 1. The first route (left side) 1→ 4 → 2 → 5 → 6 → 3 → 1 with the total length of 62 km, is a relevant selection but is not the best solution.

The second route (right side) 1 → 2 → 5 → 4 → 6 → 3 → 1 represents the must better solution as the total distance, 48 km, is less than for the first route. Suppose $c(A)$ denoted the total cost of the edges in the subset A subset of E i.e.,

$$c(A) = \sum_{u,v \text{in } A} c(u, v)$$

Moreover, the cost function satisfies the triangle inequality. That is, for all vertices u, v, w in V, w have $c(u, w) \leq c(u, v) + c(v, w)$.

Note that the TSP problem is NP-complete even if we require that the cost function satisfies the triangle inequality. This means that it is unlikely that we can find a polynomial-time algorithm for TSP.

4.1 TSP WITH THE TRIANGLE-INEQUALITY:

When the cost function satisfies the triangle inequality, we can design an approximate algorithm for TSP that returns a tour whose cost is not more than twice the cost of an optimal tour.

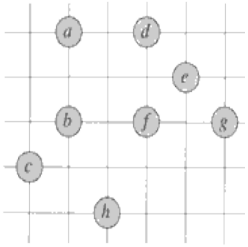
Outline of an APPROX-TSP-TOUR:

First, compute a MST (minimum spanning tree) whose weight is a lower bound on the length of an optimal TSP tour. Then, use MST to build a tour whose cost is no more than twice that of MST's weight as long as the cost function satisfies triangle inequality.

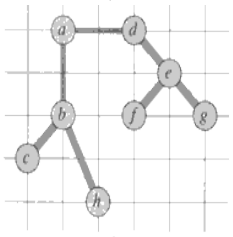
Operation of APPROX-TSP-TOUR:

Let root r be a in following given set of points (graph).

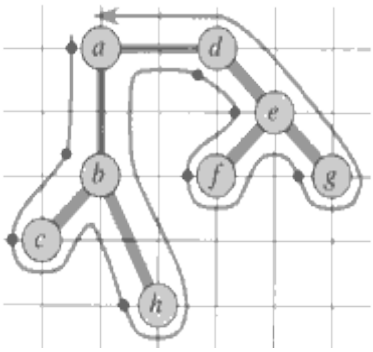
ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal



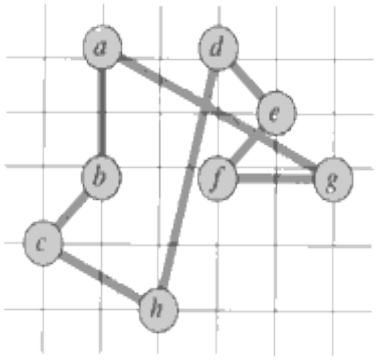
Construct MST from root a using MST-PRIM (*G*, *c*, *r*).



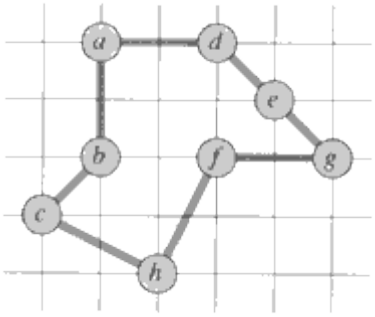
List vertices visited in preorder walk. *L* = {*a*, *b*, *c*, *h*, *d*, *e*, *f*, *g*}



Return Hamiltonian cycle.



Optimal TSP tour for a given problem (graph) would be



which is about 23% shorter.

Theorem: APPROX-TSP-TOUR is a polynomial-time 2-approximation algorithm for TSP with triangle inequality.

Proof:

1. We have already shown that APPROX-TSP-TOUR-time.

2. Let H^* denote the optimal tour. Observe that a TSP with one edge removed is a spanning tree (not necessarily MST).

It implies that the weight of the MST T is in lower bound on the cost of an optimal tour.

$$c(T) = c(H^*)$$

A "Full" walk, W, traverse every edge of MST, T, exactly twice. That is,

$$c(w) = 2c(T)$$

which means

$$c(w) \leq 2c(H^*)$$

and we have

$$c(w)/c(H^*) \leq p(n) = 2$$

That is, the cost of walk, $c(w)$, of the solution produced by the algorithm is within a factor of $p(n)=2$ of the cost $c(H^*)$ of an optimal solution.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

4.2 THE GENERAL TSP:

Without the triangle inequality, a polynomial time approximate algorithm with constant approximation ratio not exist unless *P=NP*.

Theorem: If P ≠ NP, there is no polynomial-time algorithm with approximation ratio p≥1 for general *TSP*.

Proof:

Suppose to the contrary that there exists a polynomial-time algorithm *A* for *p*≥1.

Now, use algorithm A to solve instance of HAM-CYCLE problem in polynomial-time. Since HAM-CYCLE problem is NP-complete then by theorem

If any NP-complete problem is polynomial-time solvable, then P=NP. Equivalently, if any problem in NP is not polynomial-time solvable than no NP-complete problem is polynomial solvable.

Solving HAM-CYCLE in polynomial-time implies that *P=NP*.

Since the HAM-CYCLE is NP-complete and we assumed and we assumed that *P≠NP*, a contradiction is arise.

Reduction:

Let $G = (V, E)$ be an instance of Hamiltonian cycle problem. We want to determine efficiently whether G contains a Hamiltonian cycle by making use of an algorithm A. We transform G into an instance of the TSP problem as follows:

Consider a complete graph $G`=(V, E`)$ where $E` = \{(u, v): u, v \text{ in } V \text{ and } u \neq v\}$. Assign an integer cost to each edge in $E`$ as follows:

$$c(u,v) = \begin{cases} 1 & \text{if } (u, v) \text{ in } E \\ P|V| + 1 & \text{otherwise} \end{cases}$$

Because algorithm A is guaranteed to return a tour of cost no more than P times the cost of an optimal tour, if graph G contains a Hamiltonian cycle, then A must return it. If G has no Hamiltonian cycle, then A returns a tour of cost more than $P|V|$. Therefore, we can use algorithm A to solve Hamiltonian cycle problem in polynomial time and this is impossible unless *P=NP*.

Example: Heuristic algorithm for the Traveling Salesman Problem (T.S.P) .

This is one of the most known problems ,and is often called as a difficult problem. A salesman must visit n cities, passing through each city only once, beginning from one of them which is considered as his base, and returning to it.

The cost of the transportation among the cities (whichever combination possible) is given. The program of the journey is requested, that is the order of visiting the cities in such a way that the cost is the minimum.

Let's number the cities from 1 to n ,and let city 1 be the city-base of the salesman. Also let's assume that c(i,j) is the visiting cost from i to j. There can be c(i,j)<>c(j,i).Apparently all the possible solutions are (n-1)!.Someone could probably determine them systematically, find the cost for each and every one of these solutions and finally keep the one with the minimum cost. These requires at least (n-1)! steps.

f for example there were 21 cities the steps required are (n-1)!=(21-1)!=20! steps. If every step required a msec we would need about 770 centuries of calculations.Apparently,the exhausting examination of all possible solutions is out of the question.Since we are not aware of any other quick algorithm that finds a best solution we will use a heuristic algorithm. According to this algorithm whenever the salesman is in town i he chooses as his next city,the city j for which the c(i,j) cost,is the minimum among all c(i,k) costs, where k are the pointers of the city the salesman has not visited yet (Applegate, D., Bixby, R., Chvatal, V., and Cook, W., 1998).

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

There is also a simple rule just in case more than one cities give the minimum cost,for example in such a case the city with the smaller k will be chosen.This is a greedy algorithm which selects in every step the cheapest visit and does not care whether this will lead to a wrong result or not.

4.3 TSP IN C-LANGUAGE
PSEUDO CODE:

Input: Number of cities n and array of costs c(i,j) i,j=1,..n (We begin from city number 1)
Output: Vector of cities and total cost.

- (* starting values *)
- C=0
- cost=0
- visits=0
- e=1 (*e=pointer of the visited city)
- (* determination of round and cost)
- for r=1 to n-1 do
    - choose of pointer j with
    - minimum=c(e,j)=min{c(e,k);visits(k)= 0 and k=1,..,n}
    - cost=cost+minimum
    - e=j
    - C(r)=j
- end r-loop
- C(n)=1
- cost=cost+c(e,1)

We can find situations in which the TSP algorithm doesn't give the best solution.We can also succeed on improving the algorithm.For example we can apply the algorithm t times for t different cities and keep the best round every time.We can also unbend the greeding in such a way to reduce the algorithm problem, that is there is no room for choosing cheep sides at the end of algorithm because the cheapest sides have been exhausted.

CONCLUSION

In the Lexisearch the structure of the search algorithm does not require huge dynamic memory during execution. Hence, particularly for larger problems Lexisearch appears to have relatively smaller space complexity. Also, 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. Till 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 described a search based algorithm for finding optimal solution to the Travelling Salesperson Problem. 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.

REFERENCE:

- Aiex. R. M., Resende, M. G. C., Pardalos, P. M. and Toraldo G. (2005): Grasp with Path Relinking for Three-Index Assignment. INFORMS J. On Computing, 17(2): 224–247.
- Adaptive Ant Colony Optimization for the Traveling Salesman Problem, Michael Maur (2009).
- Bhavani & Sundara Murthy, M. (2006): Truncated M-Travelling Salesmen Problem, OPSEARCH, Vol.43, No. 2.
- Boyd, A.B., 2002. Discrete mathematics topics in the secondary school curriculum.
- De Maio, A &Roveda, C. (1971): An all 0-1 Algorithm for a Certain Class of Transportation Problems, Op. Res.19, pp 1406-1418.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

- Gavish, B., Srikanth, K, 1986. An optimal solution method for large-scale multiple traveling salesmen problems. Operations research, *34(5), 698-717.*
- Huang, G. and Lim A. (2006): A Hybrid Genetic Algorithm for the Three-Index Assignment Problem. European Journal of Operational Research, 172(1):249–257.
- Karapetyan, D., Gutin, G. and Goldengorin, B. (2008): Empirical Evaluation of Construction Heuristics for the Multidimensional Assignment Problem. In: London Algorithmics 2008: Theory and Practice.
- Lin, S., Kernighan, B.W., 1973. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2), 498-516.
- P.Rama Murthy-" Operations Research linear programming"-e books.
- Ross, G.T. &Soland, R.M. (1975): An Algorithm for the Generalized Assignment Problem. Mathematical Programming; pp 91-103.
- Shapiro, J. F., 1989. Convergent duality for the traveling salesman problem. *Operations research center*, 1-14.
- 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.
- The Advantage of Intelligent Algorithms for TSP, Traveling Salesman Problem, Theory and Applications, Prof. Donald Davendra and Yuan-bin Mo (2010).