iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: III Month of publication: March 2018

DOI: http://doi.org/10.22214/ijraset.2018.3368

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Traveling Sales Man Problem Using Asp .Net

Hattarki Pooja[1]
[1]*Dep. of Computer Science, Appa Institute Of Engineering and Technology Gulbarga, Karnataka, India*

Abstract: The aim of the work is to solve the traveling salesman's problem Let us consider that there are around 300-500 cities that a sale's man needs to travel. The cities are randomly distributed at various positions. Now an algorithm must be develop to derive a smallest route, which should define both the starting point and any of the intermediate points till the last point so that all the cities are covered and that too with the smallest possible cost. The algorithm has to face 300P300 probability options. Therefore we will adopt two schemes. First an algorithm will calculate some 50 shortest path from some arbitary source and destination. These paths will be fed to a genetic algorithm, which will continue to optimize the paths. The process will continue till there is no substantial variations in the cost functions.
Keywords: TSP (travelling sales person), cities , shortest path.

I. INTRODUCTION

A. Traveling Salesman Problem

If a salesman starts at point 'a', and if the distances between every pair of points are known, what is the shortest route the salesman could take to visit all the other points and return to point 'a'?

B. Exact Algorithms

The most direct solution would be to try all the permutations (ordered combinations) and see which one is cheapest (using brute force search), but given that the number of permutations is n! (the factorial of the number of cities, n), this solution rapidly becomes impractical. Using the techniques of dynamic programming, one can solve the problem in time $O(n^2 2^n)$[4]. Although this is exponential, it is still much better than O(n!). Other approaches include:

1) Various branch-and-bound algorithms, which can be used to process TSPs containing 40-60 cities.
2) Progressive improvement algorithms which use techniques reminiscent of linear programming. Works well for up to 200 cities.
3) Implementations of branch-and-bound and problem-specific cut generation; this is the method of choice for solving large instances. This approach holds the current record, solving an instance with 85,900 cities, see below.[5]

An exact solution for 15,112 German towns from TSPLIB was found in 2001 using the cutting-plane method proposed by George Dantzig, Ray Fulkerson, and Selmer Johnson in 1954, based on linear programming. The computations were performed on a network of 110 processors located at Rice University and Princeton University (see the Princeton external link). The total computation time was equivalent to 22.6 years on a single 500 MHz Alpha processor. In May 2004, the travelling salesman problem of visiting all 24,978 towns in Sweden was solved: a tour of length approximately 72,500 kilometers was found and it was proven that no shorter tour exists.[6] In March 2005, the travelling salesman problem of visiting all 33,810 points in a circuit board was solved using CONCORDE: a tour of length 66,048,945 units was found and it was proven that no shorter tour exists. The computation took approximately 15.7 CPU years (Cook et al. 2006). In April 2006 an instance with 85,900 points was solved using CONCORDE, taking over 136 CPU years, see the book by Applegate et al [2006] [7].

C. Heuristics

Various approximation algorithms, which quickly yield good solutions with high probability, have been devised. Modern methods can find solutions for extremely large problems (millions of cities) within a reasonable time which are with a high probability just 2-3% away from the optimal solution. Several categories of heuristics are recognized.

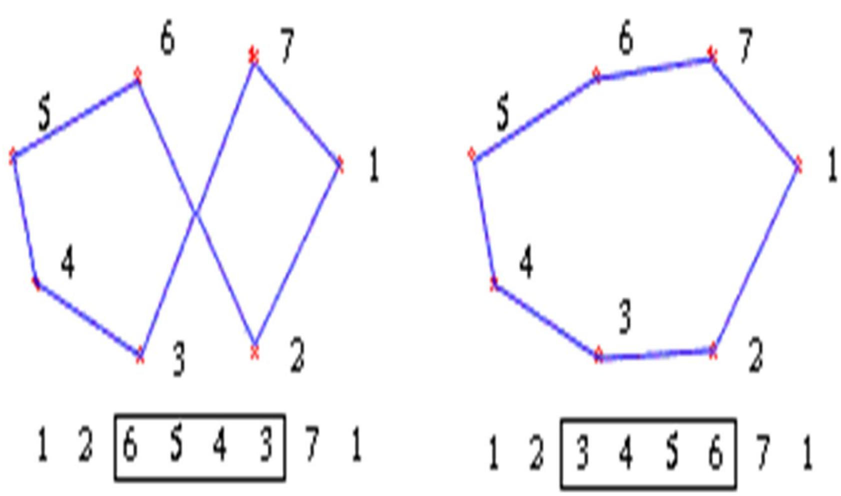
D. Constructive Heuristics

The nearest neighbour (NN) algorithm (the so-called greedy algorithm is similar, but slightly different) lets the salesman start from any one city and choose the nearest city not visited yet to be his next visit. This algorithm quickly yields an effectively short route. Rosenkrantz et al. [1977] showed that the NN algorithm has the approximation factor $\Theta(\log |V|)$ for instances satisfying the triangle inequality. And the result is always of length <= 0.5*(log(n)+1), where n is the number of cities (Levitin, 2003). For each n>1, there exist infinitely many examples for which the NN (greedy algorithm) gives the longest possible route (Gutin, Yeo, and Zverovich, 2002). This is true for both asymmetric and symmetric TSPs (Gutin and Yeo, 2007). Recently a new constructive heuristic, Match Twice and Stitch (MTS) (Kahng, Reda 2004 [8]), is proposed. MTS has been shown to empirically outperform all existing tour construction heuristics. MTS performs two sequential matchings, where the second matching is executed after deleting all the edges of the first matching, to yield a set of cycles. The cycles are then stitched to produce the final tour.

E. Example Letting The Inversion Operator Find A Good Solution

Suppose that the number of towns is sixty. For a random search process, this is like having a deck of cards numbered 1, 2, 3, ... 59, 60 where the number of permutations is of the same order of magnitude as the total number of atoms in the known universe. If the hometown is not counted the number of possible tours becomes 60*59*58*...*4*3 (about 10^{80}, i. e. a 1 followed by 80 zeros). Suppose that the salesman does not have a map showing the location of the towns, but only a deck of numbered cards, which he may permute, put in a card reader - as in early computers - and let the computer calculate the length of the tour. The probability to find the shortest tour by random permutation is about one in 10^{80} so, it will never happen. So, should he give up? No, by no means, evolution may be of great help to him; at least if it could be simulated on his computer. The natural evolution uses an inversion operator, which - in principle - is extremely well suited for finding good solutions to the problem. A part of the card deck (DNA) - chosen at random - is taken out, turned in opposite direction and put back in the deck again like in the figure below with 6 towns. The hometown (nr 1) is not counted. If this inversion takes place where the tour happens to have a loop, then the loop is opened and the salesman is guaranteed a shorter tour. The probability that this will happen is greater than 1/(60*60) for any loop if we have 60 towns, so, in a circuit with one million card decks it might happen 1000000/3600 = 277 times that a loop will disappear.

This has been simulated with a circuit of 180 card decks, from which 60 decks are selected in every generation. The figure below shows a random tour at start



In a special case when all towns are equidistantly placed along a circle, the optimal solution is found when all loops have been removed. This means that this simple random search is able to find one optimal tour out of as many as 10^{80}. See also Goldberg, 1989.

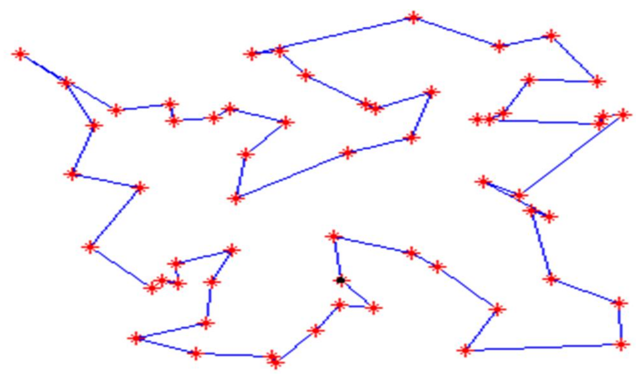


Fig: Ant colony optimization

Artificial intelligence researcher Marco Dorigo described in 1997 a method of heuristically generating "good solutions" to the TSP using a simulation of an ant colony called ACS.[9] It uses some of the same ideas used by real ants to find short paths between food sources and their nest, an emergent behavior resulting from each ant's preference to follow trail pheromones deposited by other ants. ACS sends out a large number of virtual ant agents to explore many possible routes on the map. Each ant probabilistically chooses the next city to visit based on a heuristic combining the distance to the city and the amount of virtual pheromone deposited on the edge to the city. The ants explore, depositing pheromone on each edge that they cross, until they have all completed a tour. At this point the ant which completed the shortest tour deposits virtual pheromone along its complete tour route (global trail updating). The amount of pheromone deposited is inversely proportional to the tour length; the shorter the tour, the more it deposits.

II. SYSTEM ANALYSIS

A. Existing System

The existing system determines the un approximate path for the travelling sales person Many quick algorithms yield approximate TSP solution for large city number. To have an idea of the precision of an approximation, one should measure the resulted path length and compare it to the exact path length. To find out the exact path length, there are 3 approaches, but none have been proved to be optimum.

B. Proposed System

The proposed system determines the approximate result for travelling the cities and the distance covered. The TSP, in particular the Euclidean variant of the problem, has attracted the attention of researchers in cognitive psychology. It is observed that humans are able to produce good quality solutions quickly. The first issue of the Journal of Problem Solving is devoted to the topic of human performance on TSP.

III. SYSTEM DESIGN

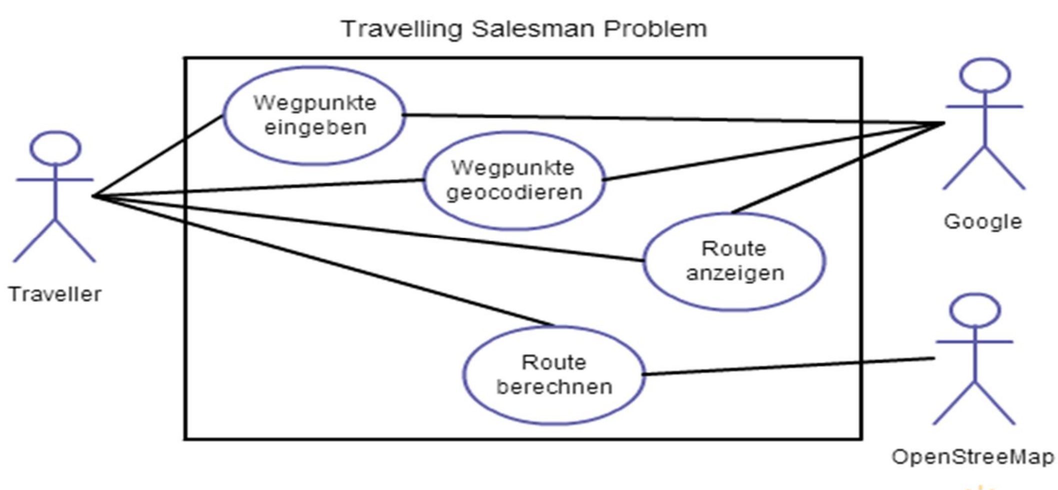


Fig: use case diagram for TSP

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

IV. SYSTEM IMPLEMENTATION

Implementation Planning System implementation is a process of making newly designed system fully operational. The system is implemented after careful testing.

A. *The following steps have been followed in he implementation of this system.*

1) System conversion.
2) User training.

Initially a primary implementation plan is prepared to schedule & manage many different activities that must be completed for a successful system implementation.

B. *A complete Implementation plan Includes The Following Items*

1) System training plan.
2) System test plan.
3) System conversion plan.
4) Overall implementation plan.

C. *System Conversion*

The system implementation could be done in the following methods.

1) Direct conversion method.
2) Parallel conversion method.
3) Phase-in method.

The system is implemented by using the parallel conversion & fade-in method. Parallel conversion is in which both new & old system is run parallel for a particular period. In the phase-in method, the new system is implemented module by module. The training should include everyone associated with the implementation, use, operation or maintenance of new system. The staff should be comfortable with the system. Coaching is a useful way of developing people's skills and abilities, and of boosting performance. It can also help deal with issues and challenges before they become major problems.

V. RESULTS

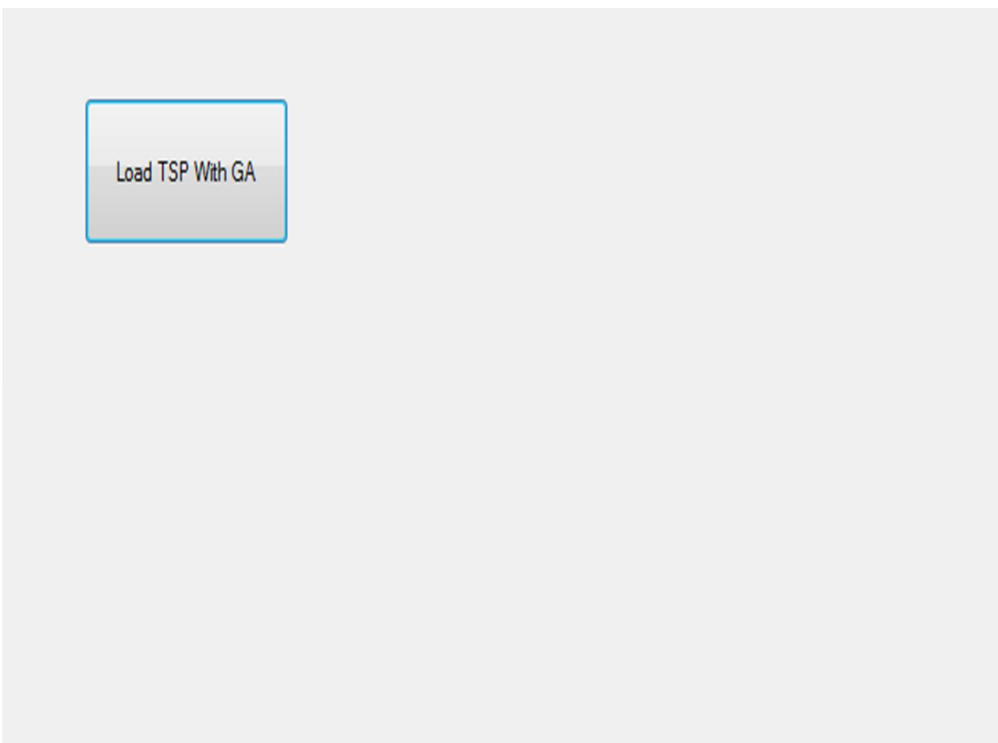


Fig: loading of TSP

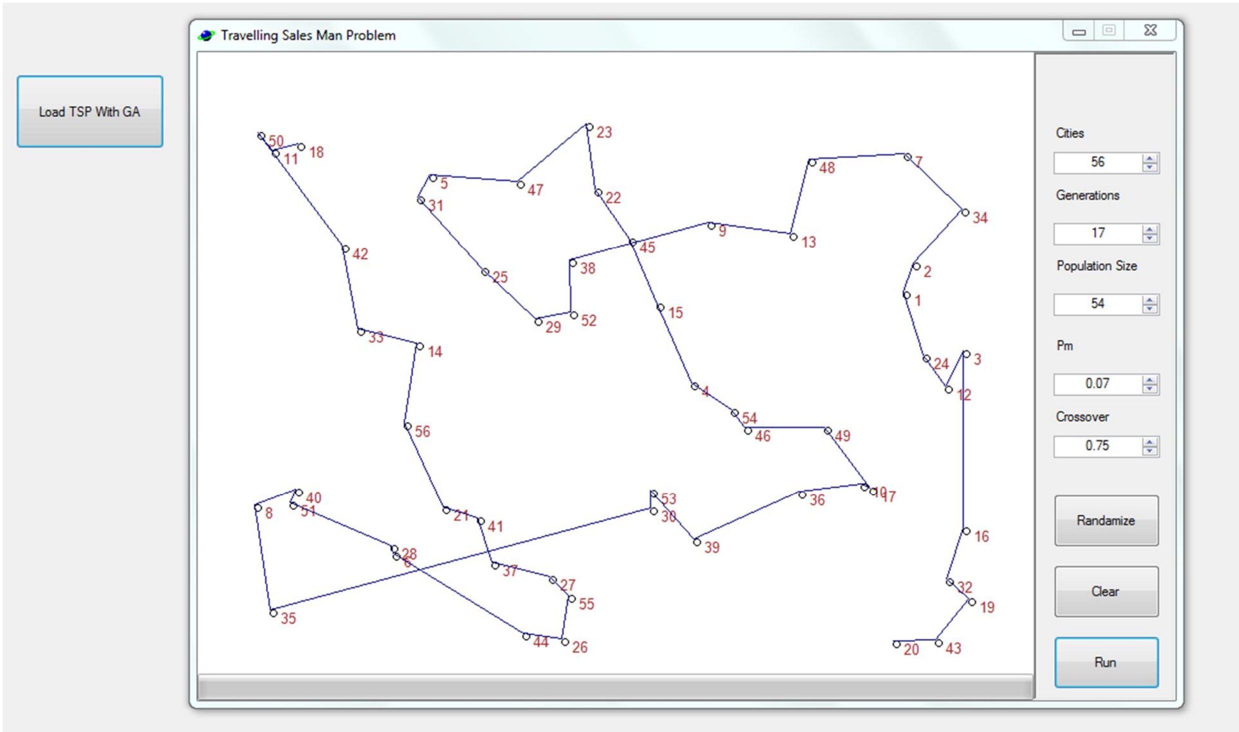


Fig: Traversing of all locations in TSP

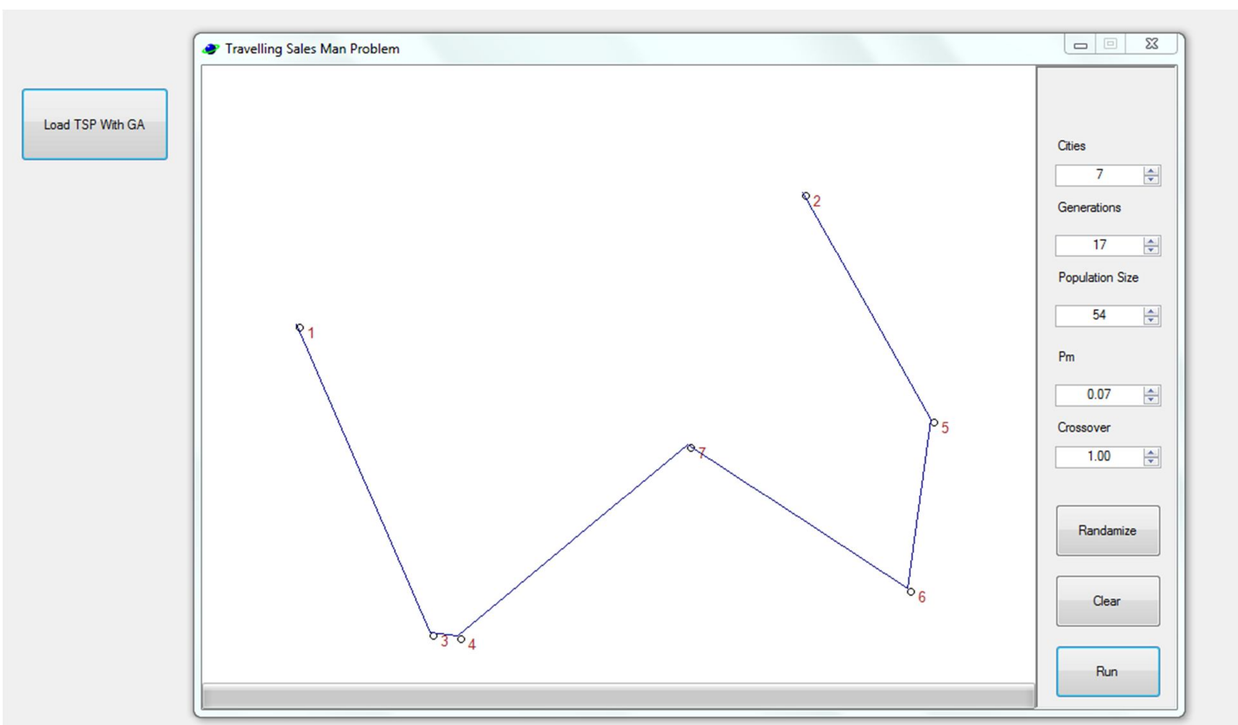


Fig: Traversing of all 5 Locations in TSP

VI. CONCLUSION

We have used Hamilton's Circuit to easily solve the complicated traveling salesman problem. A traveling salesman problem can be used to solve complicated traffic management or network routing problems. We give a brief of the technique adopted for solving the problem here.

Our search domain will be defined as a rectangular area of 500x500 entries, ranging from 0 to 500in both directions (x,y). Each pair (x,y) possible in this domain represents the position of a point. The inheriting and crossing functions will both use a pool of free indices from which used indices will be removed to keep the uniqueness of the transferred indices. Thus the developed project is near optimum solution for TSP. Main disadvantage of the current work is that is is restricted to few number of cities. We can improve the algorithm by more stronger memry management sches so that more number of cities can be incorporated in the algorithm.

REFERENCES

[1] Corman H. Thomas, Leiserson E. Charles, Rivest L. Ronald, Stein Clifford "Introduction to Algorithms" Second Edition McGrawHill Book Company
[2] Cesari Giovanni "Divide and Conquer Strategies for Parallel TSP Heuristics", Computers Operations Research , Vol.23, No.7, pp 681-694, 1996
[3] del Castillo M. Jose "A heuristic for the traveling salesman problem based on a continuous approximation", Transportation Research Part B33 (1999) 123-152
[4] Gutin Gregory,eo Anders, Zverovich Alexey "Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP" , Discrete Applied Mathematics 117 (2002), 81-86
[5] van der Poort S. Edo, Libura Marek, Sierksma Gerard, vander Veen A. A. Jack "Solving the k-best traveling salesman problem", Computers & Operations Research 26 (1999) 409-425
[6] Valenzuela I. Christine, Jones J. Antonia "Estimating the Held-Karp lower bound for the geometric TSP , European Journal of Operational Research 102(1997) 157-175
[7] Zhang Weixiong "A note on the complexity of the Assymetric Traveling Salesman Problem", Operation Research Letters 20 (1997) 31-38
[8] Web sites
[9] http://www.google.com\ it is a excellent search engine use to get any information
[10] http://eee.codeguru.com\ it is used for the information and code