Volume 3, No. 5, Sept-Oct 2012
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
REVIEW ARTICLE
Available Online at www.ijarcs.info

An Analysis of various techniques to solve Travelling Salesman Problem: A Review

Gurpreet Singh*
M.Tech Student Department of
Computer Science and Engineering
DAV Instituite of Engineering and Technology
144008, Punjab, India
gps_ghotra@yahoo.com

Vinay Chopra
Asst. Prof. Department of
Computer Science and Engineering
DAV Institute of Engineering and Technology
144008, Punjab, India
Vinaychopra222@yahoo.co.in

Abstract: Travelling Salesman problem is a nondeterministic polynomial time hard problem in combinatorial optimization studied in operational research and theoretical computer science. In this review paper several techniques like brute-force method, nearest neighbour, branch and bound, dijkstra shortest path algorithm, bellman ford, Floyd warshall algorithm and some heuristic techniques like ant colony optimization and genetic algorithm used to solve the travelling salesman problem are analysed. The comparison between these techniques is accomplished to state which is the better one for solving travelling salesman problem.

Keywords: Travelling Salesman Problem (TSP), Shortest Path Algorithm, Hamiltonian cycle Ant Colony Optimization (ACO), Genetic Algorithm (GA), Pheromone, Particle Swarm Optimization (PSO)

I. INTRODUCTION

Travelling Salesman Problem (TSP) refers to a salesman who wants to find a shortest possible tour that visit each city exactly once and return back to the starting city. In term of graph theory given n nodes (cities) and distance between each city, the goal is to find the minimum cost Hamiltonian cycle, which is a shortest tour that visit each node exactly once and return to starting node. Even though the problem is computationally difficult[1], a large number of heuristic and exact methods are known, so that some instances with tens of thousands of cities can be solved. The TSP is a classical NP-hard problem. It is one of the most studied discrete optimization problem. TSP has many variations and a large number of applications.

A number of techniques have been implemented to find the optimal and near-optimal path. Some of the approaches to solve the TSP are briefly discussed in section 2 and also, compares the results of the work done by various researchers on TSP. conclusion and future scope are given in section 3 and 4 respectively.

II. COMPARATIVE ANALYSIS OF TECHNIQUES FOR TSP

Researchers are still trying to explore the new ways which are more efficient to find the shortest route for a travelling salesman. Solving a TSP requires examining all possible Hamiltonian cycle through graph. If there are n nodes in a graph then circuits that need to be examined are **(n-1)! /2**, since a Hamiltonian circuit can be travelled in reverse order. A tour in a graph is a cycle that includes every vertex exactly once. The length of a tour is the sum of the edges weight for the edges on the tour. The tour (Hamiltonian cycle) with minimum cost is selected, which is the shortest optimal path for TSP.

Brute-Force method simply generates all possible tours and computes their distance. The shortest tour is thus optimal. Table 1 shows the estimated time it could take to solve TSP using Brute-Force method.

Table: 1 Estimated time to solve TSP using Brute Force mettod. [2]

Nodes	*No. of tours*	*Estimated time*
15	43,589,145,600	50 days
16	653,837,184,000	2 years
20	6.082255×10^{16}	193,000 years

Brute-Force method always works if given enough time and care. Because of its nature, it is convenient for relatively small number of nodes. Form table 1 it is seen that Brute-Force method is very inefficient as n (number of nodes) gets larger.

Heuristics are methods which cannot guarantee to produce optimal solutions, but which, we hope, produces fairly good solution [3]. In **Nearest neighbour heuristic** we look at all arcs coming out of the nodes (city) that have not been visited and choose the next closest city, then return to the starting city when all other cities are visited. This technique works but it does not necessarily give you the best solution as Brute-Force method. **Branch and Bound** strategy divides a problem into number of sub-problem. It is a system for solving a sequence of sub-problems each of which may have multiple possible solutions and where the solution chosen for one sub-problem may affect the possible solutions of later sub-problems. To avoid the complete calculation of all partial trees, we first try to find a practical solution and note its value as an upper bound for the optimum. As the distance exceeds the distance of the upper bound the calculations are done. If a new cheaper solution was found, its value is used as the new upper bound. This method is convenient for 40 to 60 nodes (cities). The Brute-Force and Branch and Bound methods are implemented in MATLAB with the code written by Dr.TimothyRedl and result shows that branch and bound methods solves the problem fast.

Dijkstra algorithm is a graph search algorithm that solves the single-source shortest path problem for a graph with non- negative edge path cost[4]. Dijkstra algorithm returns a shortest path tree, containing the shortest path from a stating vertex to each other vertex, but not necessarily the

© 2010, IJARCS All Rights Reserved

shortest paths between the other vertices, or a shortest route that visits all the vertices. The TSP problem is not finding the shortest way between two points, but in making a route between all the points which are optimal. Dijkstra algorithm do not satisfies TSP requirement-either not return to the starting city, or repeat cities.**Bellman ford algorithm** can be used to solve the single source shortest path problems in which edge weight may be negative. This algorithm returns a Boolean value which indicates whether there is negative weight cycle or not in a particular graph. If there is a negative weight cycle which is reachable from the source vertex, then Bellman Ford algorithm indicates that there is no any solution but if there is negative cycle then the algorithm produces the shortest path from the single source vertex to the remaining vertices. If G(V,E) be the graph, Where V represents the set of vertices & E represents the set of edges, then the time complexity for Bellman Ford algorithm is O(|V||E|)[5]. The original version of dijkstra's algorithm ran in $O(V^2+E)$. The complexity has been reduced to $O(E+V\log^2 V)$ using Fibonacci heaps.All pair of shortest path problem can be solved by **Floyd Warshall algorithm** within the time complexity of $O(V^3)$. But, the constraint is that there is no any negative weight cycle in the graph but the edges may be of negative weight. **Johnson algorithm** solve the problem within the time complexity of $O(V^2\log V +VE)$. If the graph contains negative cycle it will report that graph contains a negative cycle otherwise it returns a particular matrix which shows the shortest distance among the vertices. The complexity of existing Floyd warshall algorithm is reduced form $O(n^3)$ to $O(n^{3-\varepsilon})$ using proposed Floyd warshall algorithm[6].

Ant Colony Optimization (ACO), introduced by Marco Dorigo is a probabilistic technique for solving computational hard problems which can be reduced to finding good paths through graphs. ACO is inspired by the behaviour of ants in finding optimal paths from the colony to food. In the real world, ants (initially) wander randomly, and upon finding food return to their colony while laying down trails of a special chemical named “pheromone”. If other ants find such a path (pheromone trail), they are likely not to keep traveling at random, but instead follow the trail, returning and reinforcing (with pheromone) it if they eventually find food. ACO can effectively solve the TSP with near optimal solution with low number of iteration and in comparatively short time. The quality of solution depends on the number of ants. The lower no. of ants allows the individual to change the path much faster. The higher number of ants in population causes the higher accumulation of pheromone on edges, and thus an individual keeps the path with higher concentration of pheromone with a high probability[7]. Table 2 summarises the result of simulation by taking different number of ants over 1000 iterations.

Table 2: Summary of result of ACO simulation [7]

Iter	*m*	*k*	*L*
1000	100	1713	10
1000	1000	1612	34
1000	5000	1532	21
1000	10000	1465	242

Where m is the number of ants, k is tour length, l is iteration at which problem is solved. So with higher number of ants the solution is more optimal than the solution with lower number of ants. The ACO variants like Ant System (AS), Elitist Ant System (EAS), Rank-based Ant System (AS_{rank}), Min-Max based Ant System (MMAS) can be used for more complex optimization. These variants differ from each other in term of either the way of route construction or pheromone update scheme[8]. Out of all these variant EAS has fastest convergence at local optima. So this is better algorithm as compared to other. If optimum course is taken into consideration then RAS will be the most optimized ACO algorithm.

III. CONCLUSION

After studying all the techniques to solve TSP, it is concluded that the traditional algorithms have major shortcomings: firstly they are not suitable for negative edge networks; secondly they exhibits high computational complexity. Therefore, using developed ACO algorithm, a group of ants can effectively explore the graph and generate the optimal solution. Although, researchers have got remarkable success in designing a better algorithm in term of space and time complexity to solve shortest path problem. But ACO for TSP efficient and can solve the problem in short time. The performance of ACO algorithm depends on the appropriate setting of parameters. These parameters depend on the problem instance in hand and also on the required solution accuracy.

IV. FUTURE SCOPE

Techniques like genetic algorithm (GA), particle swarm intelligence can be explored and combined with ACO to enhance the performance of algorithm in finding the optimal solution quickly. ACO can be improved by embedding local optimization heuristic like 2-opt, 3-opt or Lin-Kernighan. In software testing, ACO can be used to generate optimal test cases, leading to reduction in timing for testing. ACO can be used in MANET to find the optimal route over the network.

V. REFERENCES

[1] Aplegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J.” The traveling salesman problem: a computational study”, Princeton University Press, ISBN 0-691-12993-2, 2007

[2] Calgor, H. "The Brute Force Algorithm". 24 January 2010 http://www.ctl.ua.edu /math103/ hamilton/HCalgor.htm#THE BRUTE FORCE ALGORITHM

[3] Cook, William J., William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver. Combinatorial Optimization. New York, John Wiley & Sons, 1998.

[4] Dijkstra, E.W. ”A note on two problems in connexion with graphs”. NumerischeMathematik ,1959, pp:269-271. doi:10.1007/BF01386390

[5] Jindal Pawan, Kumar Amit , KummerShishir ”Analysis of shortest path algorithms”, Global Journal of Computer Science and Technology, Vol. 10 Issue 8 Ver. 1.0 September 2010 pp 17-19.

[6] GargHimanshu, RawatParamjeet, “An Improved Algorithm for Finding All Pair Shortest Path”, International Journal of Computer Applications (0975 – 8887) Volume 47– No.25, June 2012, pp: 35-37.

[7] Ivan BrezinaJr.,ZuzanaČičková, “Solving the Travelling Salesman Problem Using the Ant Colony Optimization”, Management Information Systems, Vol. 6 , No. 4, 2011,pp: 010-014

[8] Vineet Chaudhary[1] , Parveen Kumar Yadav[2] , Pawan Kumar Dahiya[3]” Evaluation of Cost Optimization for Travelling Salesman Problem (TSP) by Comparing Various Ant Colony Optimization (ACO) Algorithms”, IJCEM International Journal of Computational Engineering & Management, Vol. 15, Issue 5, 2012, pp:38-44

© 2010, IJARCS All Rights Reserved