Literature Review

Joowon Kim

April 18, 2018

1 Literature Review

The Traveling Salesman Problem is one of the most popular method for finding the optimal solution. It was first devised by a mathematician, Karl Menger at Harvard University in 1931.[7] The problem is a combination with finding the shortest path that salesman start from certain city and visit other cities and return. TSP is a Nondeterministic Polynomial-time Complete that decision problem is in NP and NP-hard.[5] To find the answer, there are two approaches. First, the obvious solution that can be obtained by brute force. Second, approximate solution by using multiple heuristic algorithms.

To solve the TSP by brute force is accurate but the time complexity is $O(N!)$ for N is the number of cities [1]. It has inefficient time complexity because although N is a small number, $O(N!)$ makes it permutation so the time increase dramatically. To be practical, using optimized heuristic algorithm is better since there are mass amount of samples exist. Some examples for efficient heuristic algorithm are greedy algorithm,[4] genetic algorithm,[6] simulated annealing.[2] These heuristic algorithms estimate to the optimal solution with less step and it has less complexity.

Greedy algorithm has the simple way of solving TSP. First, select a random city in given N cities and set the starting city as N_0. Next, among unvisited cities, find the closest city. Than, mark the current city as visited. Repeat this until all of the cities are marked. Finally, return to N_0. In this way, it is easy to implement and effective for small sample. The complexity of this algorithm is $O(N^2 log_2(N))$ for N is the number of unvisited cities.

Genetic Algorithm is a method for searching that is inspired by the evolutionary process based on natural selection. First, they generate the possible solutions. Than, change the genes gradually to make the better solution. Gene is the solutions of the problem and they are transformed to get a better solution. The process of Genetic Algorithm has initialization, selection, crossover, mutation, replacement, loop and quit. First, in initialization step, the solution is achieved by genetic algorithm is represented by gene. Next, in selection step, the fitness of each genes are measured. The solution might be different or cannot be obtained depending on the fitness. In crossover step, use the gene that is selected and build the next gene using several methods. In mutation, use the selected genes and construct the next genes using several methods. Next step, replacement, replace the current gene with the next generation gene. In the loop step, repeat the selection, crossover, mutation and replacement until the change disappears. After these are done, check if the solution of the obtained gene is what you want and then quit.[3]

Simulated Annealing steadily move close to the answer to solve the problem. For example, let's say we need to find out the highest mountain. And assumed we do not have any background information about which mountain is the highest. In a trivial way, it would look for the place higher than current location and repeat this until the highest point is found. However, in this case, since there are many mountains, the probability of that mountain peaks to be highest is low.[8] Simulated annealing rolls a dice higher to increase the probability of getting to the highest peak. With certain probability, if a number comes out, it slips down and than climb from that point again. It might take a long time to climb to the peak but it tends to climb to the highest mountain peak.[2]

References

[1] Haider A Abdulkarim and Ibrahim F Alshammari. Comparison of algorithms for solving traveling salesman problem. Technical report, International Journal of Engineering and Advanced Technology(IJEAT), 2015.

[2] T.H. Sreenivas D.Janaki Ram and K.Ganapathy Subramaniam. Parallel simulated annealing algorithms. *Journal of Parallel and Distributed Computing*, 18:207–212, 1996.

[3] Brian J. Rosmaita Dirk Van Gucht John J. Grefenstette, Rajeev Gopal. Genetic algorithms for the traveling salesman problem. *In Proceedings of the first International Conference on Genetic Algorithms and their Applications*, pages 160–165, 1985.

[4] David S. Johnson. Local optimization and the traveling salesman problem. *Automata, languages and programming*, pages 446–461, 1990.

[5] E.L. Johnson P. Wolfe M. Held, A.J. Hoffman. Aspects of the traveling salesman problem. 28(4), 1984.

[6] Narges Mahmoodi Darani Mohammad Sedighpour, Majid Yousefikhoshbakht. An effective genetic algorithm for solving the multiple traveling salesman problem. *Journal of Optimization in Industrial Engineering*, 8:73–79, 2011.

[7] Alexander Schrijver. On the history of combinatorial optimization. *Handbooks in Operations Research and Management Science*, 12:1–68, 2005.

[8] C.D.Gelatt S.Kirkpatrick and M.P.Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.