A COMPARITIVE STUDY OF NEAREST NEIGHBOUR ALGORITHM AND GENETIC ALGORITHM IN SOLVING TRAVELLING SALESMAN PROBLEM

Ajaz Ahmed Khan [1], **Mrs. Himani Agrawal** [2]

[1] *PG Student Electronics and communication department SSGI FET Bhilai, Chhattisgarh, India*

[2] *Assistant Professor Electronics and communication department SSGI FET Bhilai, Chhattisgarh, India*

---***---

Abstract - *In this paper, we have used two algorithms, i.e. the Nearest Neighbor algorithm and Genetic Algorithm to solve the Travelling Salesman problem. The Travelling Salesman problem is a widely studied problem in computational mathematics. In the Travelling Salesman problem, a salesman has to visit all the cities first and then move to the same city from where it has started its whole journey. One of the concerns here with this problem, here is that the salesman cannot visit a single city twice. There are several optimization techniques such as ant colony optimization, particle swarm optimization, genetic algorithm, nearest neighbor algorithm, etc. out of which we have solved Travelling salesman problem using a Genetic algorithm and Nearest neighbor algorithm. In our paper, we have shown that for the same group of cities plotted at different location, Genetic Algorithm has given significantly better results than Nearest Neighbor Algorithm*

Key Words: **travelling salesman problem, genetic algorithm, nearest neighbor algorithm, optimization, particle swarm optimization, ant colony optimization**

1.INTRODUCTION

The Travelling Salesman problem is one of the most widely studied computational mathematics. The travelling salesman problem was first developed in 1930s and is used as the standard for many optimization methods. The Travelling salesman problem was used to minimize the cost of travelling expenses of a salesman by finding the shortest route. The shortest path will also save time of the salesman, but this is not under our concern. In this Travelling Salesman problem, our main aim is to find the shortest path, in which a salesman has to visit all cities and return to the same city from where it has started its journey by following the shortest path. It can also be used by a school administration to decide the route for the school bus by following the shortest path. Generally TSP is more useful in transport applications, but there are also many industrial applications such as grading a micro-controller, assigning the location of the drilling machine to drill holes in a circuit board, overhauling the gas turbine engines, order picking in the warehouse and various applications. In Fig.1, it is shown that there are five cities A, B, C, D and E and each city is connected to all other cities. If the salesman starts its journey from ‘A’, then it will visit each city B, C, D, E, and then return to ‘A’ by following the shortest path. Although in Travelling salesman problem, the salesman can visit any city in any order, but it has to visit a single city only once.

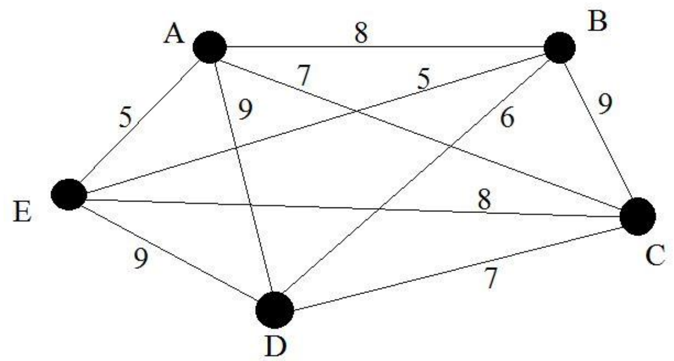

Fig.1 An Instance of Travelling Salesman problem

There are two types of Travelling Salesman problem in which the first one is a Symmetric Travelling Salesman problem and the other one is an Asymmetric Travelling Salesman problem. At Symmetric TSP, the distance between each city is same in the opposite direction as shown in Fig.2. In Asymmetric TSP, there may be or may not be pah exist between two cities. Many approximations and heuristic algorithm have been utilized to find the solution of the Travelling Salesman problem. Some of them are ant colony optimization, particle swarm optimization, genetic algorithm, nearest neighbor algorithm, etc. out of which we have utilized Genetic Algorithm and Nearest Neighbor Algorithm. In the latter part, we have explained both Nearest Neighbor Algorithm and Genetic Algorithm in Section III.

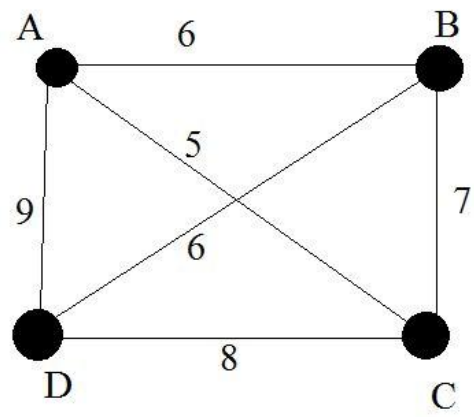

Fig.2 Symmetric Travelling Salesman problem

2. RELATED WORKS

In [2] Jinhui Yang et al. presented a paper in which Travelling Salesman problem is solved using a generalized chromosome genetic algorithm. The Generalized Chromosome Genetic Algorithm could also be used to solve the classical traveling salesman problem (CTSP), which has not been reported by others. In his paper, the generalized chromosome characteristics are analyzed and the feasibility of consistently solving the GTSP and CTSP is verified. In [4], Gohar Vahdati et al. presented a paper which proposed a new approach to Solve Traveling Salesman Problem Using Genetic Algorithm Based on Heuristic Crossover and Mutation Operator. A heuristic crossover and mutation operation have been proposed to prevent premature convergence. Presented operations try not only to solve this challenge by means of a heuristic function, but also considerably accelerate the speed of convergence by reducing excessively the number of generations. D. Kaur et al. presented a paper in which hybrid genetic algorithm used to solve Travelling Salesman problem. A novel hybrid genetic algorithm for solving a Traveling Salesman Problem (TSP) is presented based on the Nearest Neighbor heuristics and pure Genetic Algorithms (GA).The hybrid genetic algorithm exponentially derives higher quality solutions in relatively shorter time for hard combinatorial real world optimization problems such as Traveling Salesman Problem (TSP) than the pure GA.

3. METHODOLOGY

3.1 NEAREST NEIGHBOUR ALGORITHM

The Nearest Neighbor algorithm was one of the first algorithms used to find a solution for the travelling salesman problem. With this problem, the salesman starts in a random city and repeatedly visits the nearest city until all cities have been visited. It quickly finds a short tour, but usually not the best path. The nearest neighbor algorithm is very easy to put into practice and works very fast, but sometimes this algorithm can miss smaller routes which can be easily observed with human eyes, due to its "greedy" nature.

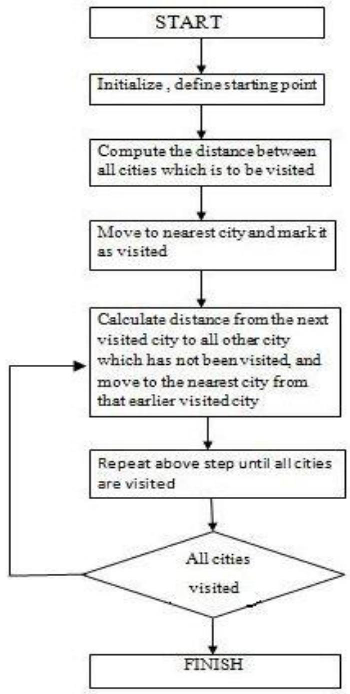

Fig.3 Flow Chart of Nearest Neighbor Algorithm

If the last few tours are almost similar to the initial tours then the solution is satisfactory and if the last few tours are very much longer then the initial ones then there may be a better solution.

Steps of algorithm:

1. Start in any random city as current city.

2. Find out the shortest path connecting current city and an unvisited city.
3. Set current city as visited city.
4. Mark V as visited.
5. If all the cities are visited, then stop.
   Go to step 2

3.2 GENETIC ALGORITHM

Genetic Algorithms (GAs) are heuristic algorithm developed from biological systems which is based on Darwinian theory, i.e. "survival of the fittest contest". According to this algorithm, the best of a particular generation, surviving while the weak one die out. Each generation contains populations of threaded like structure called chromosome that are present in our DNA. Every individual represents a particle in a search space and one of them is a possible solution. The individuals later on participate in the process of evolution.

Genetic Algorithms are generally based on the best and weak genes of chromosomes in a population of individuals using the following steps:

Individuals in a population will compete for an objective and partners or mates.

Individuals who have great outcomes in their each "competition" will yield more offspring than those individuals that perform very poorly.

Now the genes from 'better' individuals will meet together to produce an offspring which will produce even a better offspring than their own parents.

Each generation of offspring will mate together again to produce a better offspring than itself.

In this way, each successive generation will become more suited to the environment and will survive further.

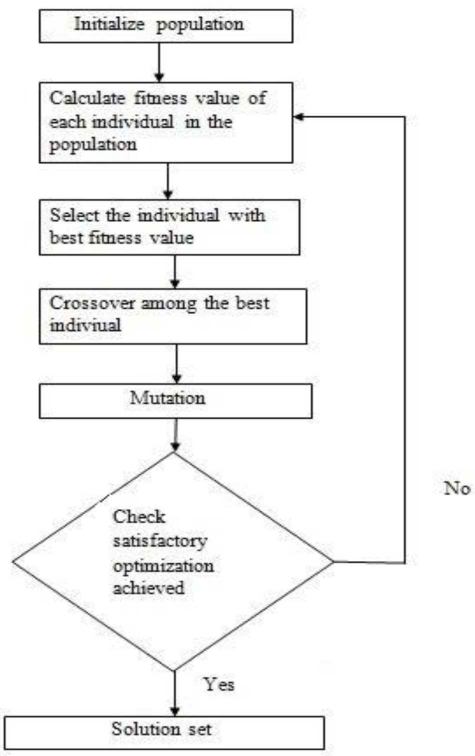

Fig.4 Flow chart of Genetic Algorithm

In our paper, we have solved the Travelling Salesman problem using Genetic Algorithm as well as the Neighbor Algorithm. Genetic Algorithm does not give you the exact shortest path, but it does provide the optimum solution. Here the optimum solution is nothing but close to the shortest path itself. Genetic Algorithm is very much useful in finding the solution of Travelling Salesman problem for small as well as a large number of cities. If a Travelling Salesman problem contains a small number of cities for e.g. 5 cities then there are 4! Possibilities i.e. 24 possibilities similarly for 10 numbers of cities, there are 9! Possibilities i.e. 362880 possibilities which are huge. Now we can say that for 20 cities problem there will be 19! Possibilities among which it is almost impossible to find the shortest path within short duration. However, it is not necessary that the solution of Genetic Algorithm is the shortest but it will be close to the shortest path if properly applied. Later on, in the RESULT section we have shown that we are getting better results with the Genetic Algorithm as compared to Nearest Neighbor Algorithm.

4. RESULT

In our paper we have located fifty cities at different location as shown in Fig.5 and distance between each of the cities is calculated by the equation given below:

$$d = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

Where 'a' is the starting city and 'b' is the last city.

We have taken the same group of cities as shown in Fig.5 in solving the Travelling Salesman problem so that it will be easy for us to compare the results of both Nearest neighbor and Genetic Algorithm.

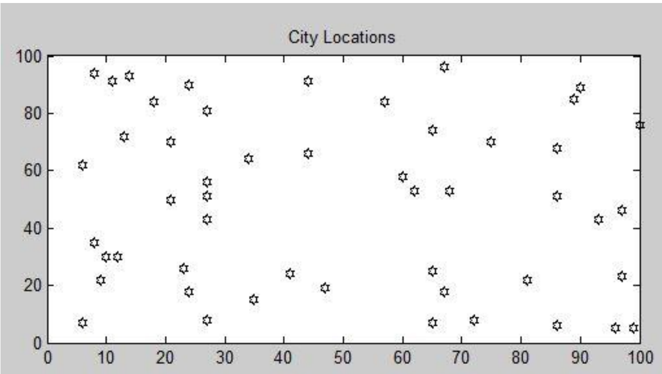

Fig.5 City Locations

In Fig.6, we have found the shortest path using Nearest Neighbor Algorithm

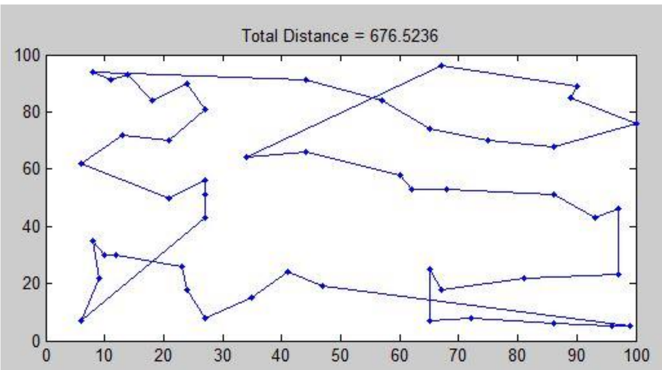

Fig.6 Shortest path using nearest Neighbor Algorithm

In Fig.7, we have found the shortest path using Genetic Algorithm.

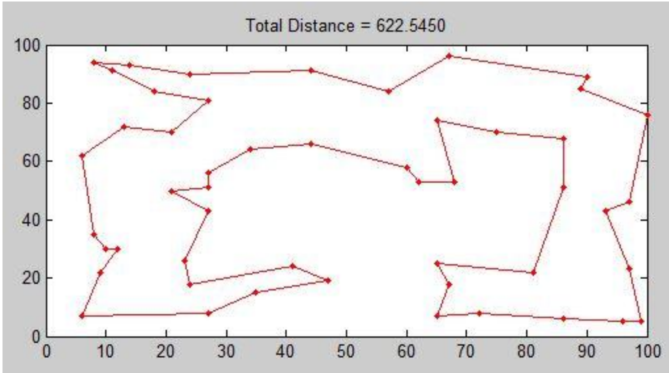

Fig.7 Shortest path using Genetic Algorithm

In Fig.5, we have shown fifty cities and for the same group of cities we have found the shortest path using a Nearest Neighbor algorithm as shown in Fig.6 and shortest path using Genetic algorithm as shown in Fig.7. From Fig.6 and Fig.7 it is clear that Genetic algorithm is finding the shortest path for a large number of cities effectively and providing better results as compared to nearest Neighbor Algorithm. In this paper, we have used Nearest Neighbor algorithm and a Genetic Algorithm to solve the Travelling Salesman problem. For a small number of cities, the Travelling Salesman problem can be easily and quickly solved by the Nearest Neighbor algorithm as compared to Genetic Algorithm. For a large number of cities, the Travelling Salesman problem can be easily solved by Genetic Algorithm as compared to a Nearest Neighbor algorithm. In Future, we would like to solve the travelling Salesman problem using Particle Swarm optimization and compare the results with Genetic Algorithm.

REFERENCES

[1] Jun Zhao, Quanli Liu, Wei Wang, Zhuoqun Wei, Peng Shi, "A parallel immune algorithm for traveling salesman problem and its application on cold rolling scheduling", J. Zhao et al. / Information Sciences 181 (2011) 1212–1223.

[2] Jinhui Yang, Chunguo Wu, Heow Pueh Lee, Yanchun Liang, "Solving traveling salesman problems using generalized chromosome genetic algorithm", Progress in Natural Science 18 (2008) 887–892 January 2008.

[3] Fischetti M, Salazar JJ, Toth P. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Oper Res 1997;45:378–94.

[4] Gohar Vahdati, Mehdi Yaghoubi, Mahdieh Poostchi, M. B. Naghibi S, "A New Approach to Solve Traveling Salesman Problem Using Genetic Algorithm Based on Heuristic Crossover and Mutation Operator", 2009 International Conference of Soft Computing and Pattern Recognition PP 112-116.

[5] Q. Zhu , S.Chen, "A New Ant Evolution Algorithm to Resolve TSP Problem", IEEE Sixth International Conference on Machine Learning and Applications (ICMLA 2007), pp. 62-66, 2007.

[6] Z. Cai, "Multi-direction Searching Ant Colony Optimization for Traveling Salesman Problems", In Proceedings of the 2008 international Conference on Computational intelligence and Security, CIS. IEEE Computer Society, Washington, DC, Vol. 02, pp.220-223, 2008

[7] Y. Xing , Z.Chen, J. Sun, L. Hu, "An Improved Adaptive Genetic Algorithm for Job-Shop Scheduling Problem", In Proceedings of the Third International Conference on Natural Computation (ICNC), IEEE Computer Society, Washington, DC, vol. 04, pp. 287-291, 2007.