Solving TSP using Genetic Algorithm and Nearest Neighbour Algorithm and their Comparison

Khushboo Arora, Samiksha Agarwal, Rohit Tanwar

Abstract— Travelling Salesman Problem is an intensively studied problem in the field of Combinatorial Optimization. Being an NP-Hard problem it is widely studied in the area of optimization. A problem is NP-Hard if its approximate solution is derived from the solution of NP problem, i.e. an algorithm that is used to solve NP problem can be modified to find the approximate solution to NP-hard problem.The main objective of TSP is to find the minimum distance by traversing each of the given set of cities atleast once and then traversing back to the start city.The paper is aimed to provide a method for solving TSP using both Genetic Algorithm and nearest neighbour Algorithm and provide efficient results. Also, this paper provides the comparison between the two algorithms based on various parameters that helps to choose the better algorithm as per the needs.

Index Terms— Combinatorial Optimization, Genetic Algorithm, Nearest Neighbour, NP-Hard, Travelling Salesman Problem.

1 Introduction

1.1The Travelling Salesman Problem

Being a combinatorial optimization problem, Travelling Salesman Problem seems to be very simple when the statement is given, but at the same time it is extremely difficult to solve[1]. In the field of computer science and operation research the problem is also known as NP-hard problem, which cannot be solved exactly in the polynomial time.

TSP can be described as: Given a set of cities, and the distances between each pair of the cities, the aim of the salesman is to find a minimum path that visits each city exactly once and gives the minimized the total distance travelled [2]. TSP can be used in number of fields such as military and traffic.

1.2 Genetic Algorithm

Genetic Algorithm was introduced by John Holland along with his colleagues and students in the mid-1970s, at the University of Michigan[6]. GA is totally based upon the "Survival of the Fittest" and "Natural Genetic" principal, in order to produce better solutions. GA operates on population of solutions. For each new generation a new set of solutions are formulated by selecting individuals according to their fitness value and then breeding them together using operators.

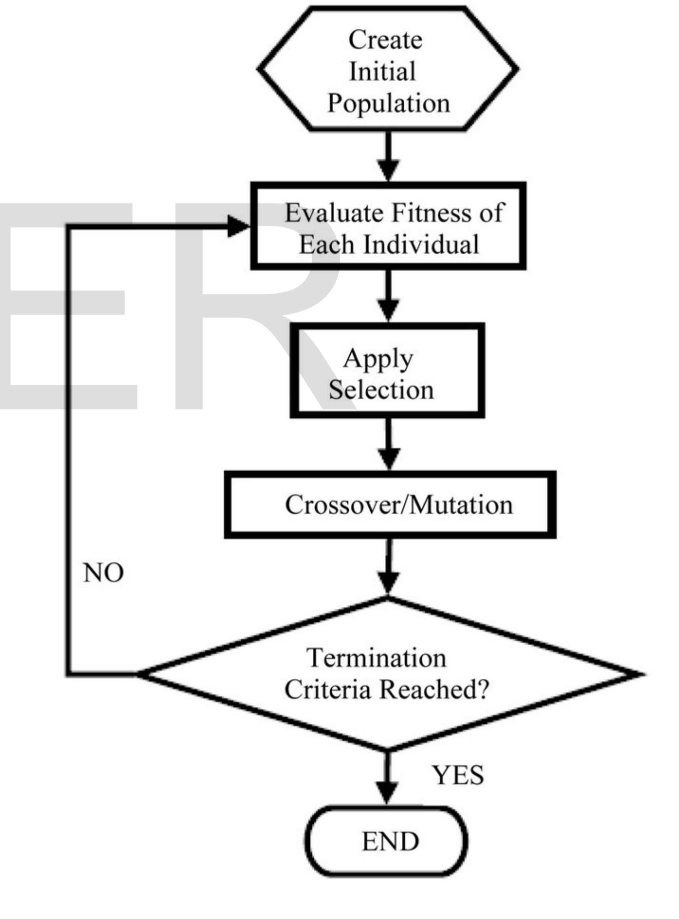

Create Initial Population
Evaluate Fitness of Each Individual
Apply Selection
Crossover/Mutation
NO
Termination Criteria Reached?
YES
END

Fig.1. Flowchart of Genetic Algorithm [3]

Algorithm of GA:

Step 1: Generate any random routes and calculate their fitness value.

Step 2: Repeat the following procedure for a given number of iteration times:

- *Khushboo Arora is currently pursuing masters degree program in Computer Engineering in Manav Rachna University,India, PH:+919711858087. E-mail: khushbooarora3@gmail.com*
- *Samiksha Agarwal is currently pursuing masters degree program in computer engineering in Manav Rachna University,India, PH:+919268571619.*
- *E-mail: samiksha1009@gmail.com*
- *Rohit Tanwa, Assistant Professor, Manav Rachna University, Faridabad, India, PH:+919992257914, E-mail:rohit@mru.edu.in*

a) Select any two best routes from the given random routes.
b) Reproduce those two routes to produce new best routes.
c) After Reproduction replace the best new routes with any two worst routes.

Step 3: Return the best route.

1.3 Nearest Neighbour Algorithm

The Nearest Neighbour Algorithm is an approximate algorithm for finding a sub-optimal solution to the TSP [3]. In this algorithm, it starts with a city as a starting city and repeatedly visits all the cities until all the cities have been visited exactly once. It finds a shortest path, but solution is not the optimal one.

The algorithm of NN is:

Step 1: Start with any random vertex, call it current vertex.

Step 2: Find an edge which gives minimum distance between the current vertex and an unvisited vertex, call it V.

Step 3: Now set that current vertex to unvisited vertex V and mark that vertex V as visited.

Step 4:Terminate the condition, if all the vertices are visited atleast once.

Step 5: Go to step 2.

This paper is organized in the following manner. Section II gives the related work that has been done in terms of travelling salesman problem, genetic algorithm and nearest neighbour algorithm. Section III describes the methodology that would be used to solve the problem. The experiment results are shown in Section IV. Section V concludes the work carried out and Section VI gives the future scope providing the options in which this work can be extended.

2 Related Work

In the work proposed by Varshika, in the travelling salesman problem using genetic algorithm [1] provides a method which is flexible for solving the TSP using genetic algorithm. They used TSP as a domain for solving NP-hard problems. Integer programming and graph theory algorithms are used to solve this problem. In order to find a maximal approximation of a specific problem with the constraint of reduced cost, the referenced paper gives an implementation using genetic algorithm.

A. Aranganayaki solves the travelling salesman problem using the heuristic method of genetic algorithms. Genetic algorithm finds good solutions for the TSP, but the solution depends how the problem is encoded and also depends on which mutation and crossover are used in the algorithm. The research introduces a binary matrix which is a new method for representing chromosomes and new fittest criteria for the crossover process. The algorithm is also applied to solve symmetric as well as the asymmetric Travelling salesman problem.

K.s. Tang and Sam Kwong proposed Genetic algorithms: Concepts and applications[6] in 1996. The paper gives an introduction about the genetic algorithm and explained when and why the Genetic algorithm used as an optimization tool. They provided the features of GA in terms of operators solved complex and conflicting problems using it.

Naveen kumar, Karambir, Rajiv Kumar provided a literature survey for solving tsp using genetic algorithm [7]. They provided number of genetic operators used in solving TSP using genetic algorithm such as crossover, mutation or combination of them. After the analysis of the survey, the researchers observed that in future new operators can be introduced to increase the performance of genetic algorithm for solving the TSP problem.

Peng Gang [8] in 2003 described multiple heuristic algorithms to solve Travelling Salesman Problem. The two operations of smallest square method and Complete 2-Opt were combined with the genetic algorithm to solve TSP. The complete 2-Opt is typically based on the 2-Opt heuristic search method which removes all the crossed edges in the tour. The main advantage of Smallest Square method that it provides the shorter edges as compared to Complete 2-Opt. Peng Gang presented a new result which combined both the operations with GA to solve TSP. There are two more operation which was also discussed in this paper, namely the Best Part Collector and deletions. To select the best among the individuals Best path Collector was used. On the other hand, Deletion removed duplicate individual from the given population.

In 1997, Rong Yang several knowledge-augmented genetic operators were proposed, that were used to guide GA for better quality of the population by not falling into local optima. Greedy crossover was applied by the algorithm that was devised by Yang. Also, advanced mutation operations based on the 2-opt and 3-opt heuristics were used [9].

O. Taiwo, O. Josiah, A. Taiwo, S. Dkhrullahi [12] proposed an implementation method of solving TSP using Nearest Insertion and Nearest Neighbour Approaches. They provided a comparison between the two stating which algorithm gives the better result and what are the flaws in the other algorithm due to which it is not able to produce the desired result. The execution time of nearest insertion algorithm is slightly less than that of nearest neighbor algorithm. But it was observed that the solution could be found in very short computational time, which helped to reach the conclusion that the use of these two algorithms gives the acceptable results which may not be optimal but are close to the optimal result.

3 Methodology

This paper solves Travelling Salesman problem using both Genetic Algorithm and Nearest Neighbour Algorithm.

3.1 Tsp Using Genetic Algorithm

Genetic Algorithm randomly generates an initial population of strings, also called as gene pool and then applying crossover followed by mutation to create new generation that is better than the previous populations.

GA process consists of the following:

Intialization
From a population of individuals, an initial population is generated.

Fitness Function
After initializing the population, fitness value for each of the individual is calculated. The fitness values are generated using a fitness function. The function provides largest and smallest values for each of the individuals. If the individual has a larger fitness value then result will be a better solution but if a smaller value is obtained then solution obtained will not be better.

Selection
The individuals which have the highest fitness value are chosen to produce the offspring. Roulette Wheel, Tournament selection, Steady state and Rank selection, are the different selection techniques through which selection operation can be done.

Crossover
Any two individuals from the given population are picked and new generation is created by recombining the two individuals.

Mutation
Mutation is considered as a simple search operator in the algorithm. When the crossover function is completed, mutation is applied to the strings so formed. Mutation function prevents the algorithm from being trapped into local minimum.

Termination
The algorithm terminates when it has executed a given number of iterations or it has reached a termination condition and then produces the best solution.

3.2 Tsp Using Nearest Neighbour Algorithm

Nearest neighbour, was the first greedy algorithm which gave a solution for the travelling salesmen problem. The algorithm was introduced by J.G. Skellam and it was continued by the F.C. Evans and P.J Clark. In nearest neighbor algorithm, we randomly choose a city as the starting city and then traverse to all the cities closest to the staring city that do not create any cycle. This process continues till all cities are visited once.

The steps of the algorithm are:

Step 1: Select any node as starting node.
Step 2: Look for the neighbours of the starting node that have not been visited yet. Choose the node which is next closest to that node having the minimum distance.

Step 3: Repeat this procedure until all the nodes in the tour have been visited exactly once.

Step 4: Check if all the nodes are visited atleast once. If yes, then return back to the starting node which will give the complete path.

Step 5: Finally calculate the total distance of the tour.

This algorithm takes input as the number of cities and coordinates as the distances for cities which is represented in the form of adjacency matrix. If the starting node is specified it continues with it, otherwise starting node is chosen at random. The algorithm gives a sequence of all the visited vertices. But the solution produced by the algorithm is not the optimal one. As the algorithm is a greedy algorithm it misses out some of the shorter routes. These shorter routes can be detected by human insight easily. So, the nearest neighbour algorithm does not give the feasible solution.

4 Experiment Results

For the implementation, we used Windows 7 laptop of Intel core I5 with 2.5GHz processor and 4 gigabytes RAM. The implementation of NN and GA algorithms is done in Matlab. To compare the effectiveness of both the algorithms, we took three instances, 25 cities, 50 cities and 100 cities to see which algorithm gives better result in a given situation.

Three runs are done on both the algorithm based on the criteria of different number of cities. For the first run, we took 25 cities by giving the x-y coordinates. The Fig.2 depicts the graph of solving TSP using nearest neighbour algorithm. Similarly the figure Fig.3 depicts the graph of solving the Travelling salesman problem using Genetic algorithm.

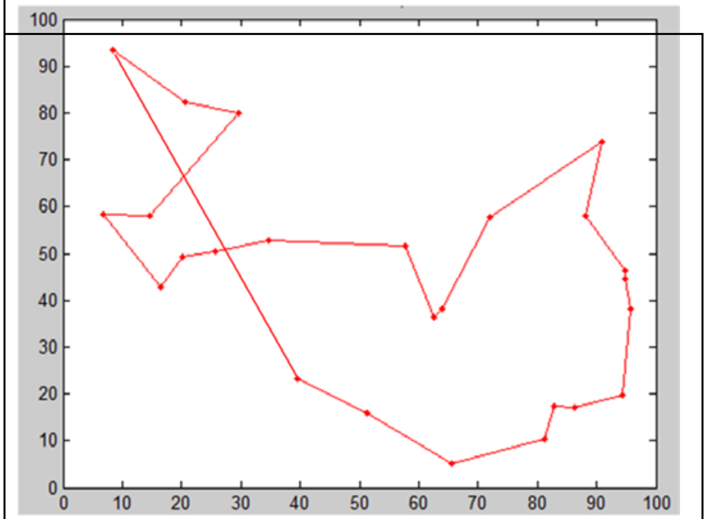


Fig2. Solving TSP using NN

Fig 3. Solving TSP using GA

Similarly we did for 50 cities and 100 cities and found out which algorithm gives the best result in all the given conditions.

The comparison is done on the basis of the total distance travelled by the algorithm and the time taken by them.

TABLE 1
COMPARISON BETWEEN NN AND GA

Algorithm	No. of Cities	Total Distance Travelled	Execution Time
Nearest Neighbour	25	42.2683	0.8631
	50	77.2339	6.6354
	100	86.4397	2.9486
Genetic Algorithm	25	41.3736	12.5727
	50	62.3458	14.5790
	100	82.9532	21.1832

RESULTS

The results in the table, helps to conclude that Genetic algorithm gives the best result with the given set of conditions.

5 Conclusion

The TSP can be solved using both the heuristic algorithms GA and NN which provides sub optimal solution. In the nearest neighbor, every node is serving as a corresponding initial node with respect to the next closet node, which concludes that nodes are not independent of one another.

The paper gives a comparison between Genetic algorithm and nearest neighbor to solve the travelling salesman problem. The comparison is based on the criteria of number of cities travelled by both the algorithms and then determines and tests the result on the basis of the distance travelled by them and the execution time.

The total distance travelled using genetic algorithm in all the three cases in less than the nearest neighbour algorithm. But the execution time of NN algorithm in all the three cases in less than the Genetic algorithm. The main aim of TSP was to find the minimum total distance travelled, so considering the main objective of TSP, it has been concluded that Genetic algorithm is considered as the best algorithm as it provides better results in terms of distance travelled.

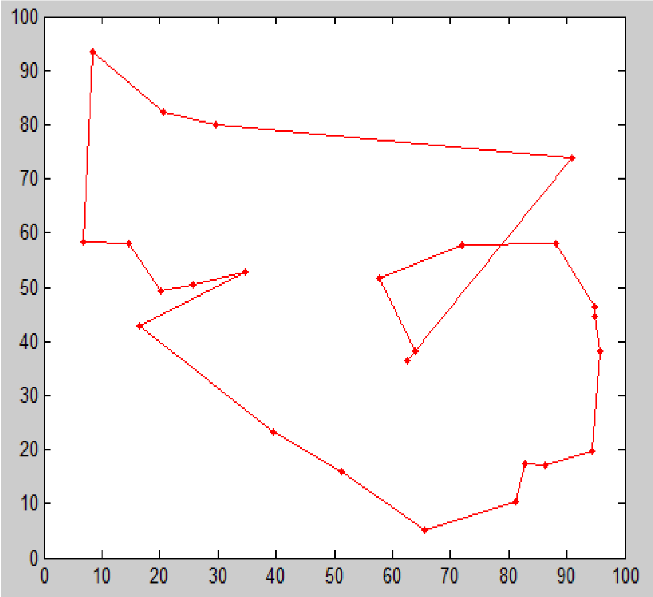
6 Future Scope

This paper presents a comparative analysis among Genetic Algorithm and Nearest Neighbour Algorithm for solving TSP and results shows that Genetic gives better results than Nearest Neighbour. But there exists many algorithms for solving TSP which can be compared in order to classify which algorithm gives the best optimal results under a given set of conditions.

In this paper we have compared on the basis of total distance travelled and execution time based on the criteria of number of cities. In future the comparison can be done on the basis of performance and cost to see which algorithm gives the better result.

Acknowledgement

We would like to thank our teacher Mr. Rohit Tanwar for providing us help and for motivating us.



REFERENCES

[1] Varshika Dwivedi, Taruna Chauhan, Sanu Saxena, Princie Agrawal, “Travelling Salesman Problem using Genetic Algorithm”, International Journal of Computer Applications, pp. 25-30

[2] Shalini Singh, Ejaz Aslam Lodhi, “Study of Variation in TSP using Genetic Algorithm and Its Operator Comparison” , International Journal of Soft Computing and Engineering (IJSCE), Vol. 3, Issue-2, May 2013, pp. 264-267

[3] http://file.scirp.org/Html/7-9601193%5Ca 9b12e 87-908f-4343-8b9d-f41cb52fad6f.jpg

[4] https://en.wikipedia.org/wiki/Nearest_neighbour_algorithm

[5] A.Aranganayaki, “Reduce Total Distance and Time Using Genetic Algorithm in Traveling Salesman Problem”, International Journal of Computer Science & Engineering Technology, Vol. 5, No. 08 Aug 2014, pp. 815-819

[6] K.s. Tang, Sam Kwong, “Genetic algorithms: Concepts and applications”, IEEE transactions on industrial electronics, Vol. 43, No. 5, Oct 1996, pp. 519-534

[7] Naveen kumar, Karambir, Rajiv Kumar, “A genetic algorithm approach to study travelling salesman problem”, Journal of Global Research in Computer Science, Volume 3, No. 3, March 2012, pp. 33-38.

[8] Peng Gang et al, “Multiple Heuristic Search Algorithm for Solving Travelling Salesman Problem‖”, IEEE, 2003.

[9] Rong Yang, “Solving Large Travelling Salesman Problems with Small Populations”. IEEE 1997.

[10] Fatma A. Karkory, Ali A. Abudalmola, “Implementation of Heuristics for Solving Travelling Salesman Problem Using Nearest Neighbour and Minimum Spanning Tree Algorithms”, International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering Vol:7, No:10, 2013, pp 987-997

[11] Chetan Chauhan, Ravindra Gupta, Kshitij Pathak, “Survey of Methods of Solving TSP along with its Implementation using Dynamic Programming Approach ”, International Journal of Computer Applications, Volume 52– No.4, August 2012, pp. 12-19

[12] Oloruntoyin Sefiu Taiwo, Ojo Josiah, Amao Taiwo, Salawudeen Dkhrullahi & Oloruntoyin Kehinde Sade, “Implementation Of Heuristics For Solving Travelling Salesman Problem Using Nearest Neighbour And Nearest Insertion Approaches”, International Journal of Advance Research, Volume 1, Issue 3, March 2013, pp. 139-155