Improving the Solution of Traveling Salesman Problem Using Genetic, Memetic Algorithm and Edge assembly Crossover

[1]Mohd. Junedul Haque
College of Computers and Info. Tech.
Taif University
Taif, Saudi Arabia

[2]Khalid. W. Magld
College of Computing and Info. Tech.
King Abdulaziz University
Taif, Saudi Arabia

Abstract— **The Traveling salesman problem (TSP) is to find a tour of a given number of cities (visiting each city exactly once) where the length of this tour is minimized. Testing every possibility for an N city tour would be N! Math additions. Genetic algorithms (GA) and Memetic algorithms (MA) are a relatively new optimization technique which can be applied to various problems, including those that are NPhard. The technique does not ensure an optimal solution, however it usually gives good approximations in a reasonable amount of time. They, therefore, would be good algorithms to try on the traveling salesman problem, one of the most famous NP-hard problems. In this paper I have proposed a algorithm to solve TSP using Genetic algorithms (GA) and Memetic algorithms (MA) with the crossover operator Edge Assembly Crossover (EAX) and also analyzed the result on different parameter like group size and mutation percentage and compared the result with other solutions.**

Keywords- NP Hard; GA(Genetic algorithms); TSP(Traveling salesman problem); MA(Memetic algorithms); EAX(Edge Assembly Crossover).

I. INTRODUCTION

The traveling salesman problem (TSP) is to find a tour of a given number of cities (visiting each city exactly once) where the length of this tour is minimized. The TSP is defined as a task of finding of the shortest Hamiltonian cycle or path in complete graph of N nodes. It is a classic example of an NP-hard problem. So, the methods of finding an optimal solution involve searching in a solution space that grows exponentially with number of city [1].

The traveling salesman problem (TSP) is one of the most widely studied NP-hard combinatorial optimization problems. Its statement is deceptively simple, and yet it remains one of the most challenging problems in Operational Research. The simple description of TSP is Give a shortest path that covers all cities along. Let G =(V; E) be a graph where V is a set of vertices and E is a set of edges. Let C = (cij) be a distance (or cost) matrix associated with E. The TSP requires determination of a minimum distance circuit (Hamiltonian circuit or cycle) passing through each vertex once and only once. And Distribution Problem, it has attracted researchers of various domains to work for its better solutions[3].

Those traditional algorithms such as Cupidity Algorithm, Dynamic Programming Algorithm, are all facing the same obstacle, which is when the problem scale N reaches to a certain degree, the so-called “Combination Explosion” will occur. A lot of algorithms have been proposed to solve TSP. Some of them (based on dynamic programming or branch and bound methods) provide the global optimum solution. Other algorithms are heuristic ones, which are much faster, but they do not guarantee the optimal solutions. The TSP was also approached by various modern heuristic methods, like simulated annealing, evolutionary algorithms and tabu search, even neural networks. In this paper, we proposed a new algorithm based on Inver-over operator, for traveling salesman problems. In the new algorithm we will use new strategies including selection operator, replace operator and some new control strategy, which have been proved to be very efficient to accelerate the converge speed.[5]

II. LITERATURE SURVEY

A. *The Traveling Salesman problem (TSP)*

The Traveling Salesman Problem (TSP) is an NP-hard problem in combinatorial optimization studied in operations research and theoretical computer science. Given a list of cities and their pair wise distances, the task is to find a shortest possible tour that visits each city exactly once [1].

The Traveling Salesman Problem (TSP) is an NP-hard problem in combinatorial optimization studied in operations research and theoretical computer science. Given a list of cities and their pair wise distances, the task is to find a shortest possible tour that visits each city exactly once [2]. With metric distances In the metric TSP, also known as delta-TSP, the intercity distances satisfy the triangle inequality. This can be understood as “no shortcuts”, in the sense that the direct connection from A to B is never longer than the detour via C. [2]

$$C_{ij} <= C_{ik} + C_{kj}$$

Exact algorithms the most direct solution would be to try all permutations (ordered combinations) and see which one is cheapest (using brute force search). The running time for this approach lies within a polynomial factor of $O(n!)$, the factorial of the number of cities, so this solution becomes impractical even for only 20 cities. One of the earliest applications of dynamic programming is an algorithm that solves the problem in time $O(n^2 2^n)$[2].

The dynamic programming solution requires exponential space. Using inclusion–exclusion, the problem can be solved in time within a polynomial factor of 2^n and polynomial space. Improving these time bounds seems to be difficult. For example, it is an open problem if there exists an exact algorithm for TSP that runs in time $O(1.9999^n)$ [2]

B. Genetic Algorithms

Genetic algorithms are an optimization technique based on natural evolution. They include the survival of the fittest idea into a search algorithm which provides a method of searching which does not need to explore every possible solution in the feasible region to obtain a good result. Genetic algorithms are based on the natural process of evolution. In nature, the fittest individuals are most likely to survive and mate; therefore the next generation should be fitter and healthier because they were bred from healthy parents. This same idea is applied to a problem by first 'guessing' solutions and then combining the fittest solutions to create a new generation of solutions which should be better than the previous generation. We also include a random mutation element to account for the occasional 'mishap' in nature [7].

Outline of the Basic Genetic Algorithm

1) *[Start] Generate random population of n chromosomes (suitable solutions for the problem)*
2) *[Fitness] Evaluate the fitness f(x) of each chromosome x in the population*
3) *[New population] Create a new population by repeating following steps until the new population is complete*
4) *[Selection] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)*
5) *[Crossover] With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.*
6) *[Mutation] With a mutation probability mutate new offspring at each locus (position in chromosome).*
7) *[Accepting] Place new offspring in a new population.*
8) *[Replace] Use new generated population for a further run of algorithm.*
9) *[Test] If the end condition is satisfied, stop, and return the best solution in current population.*
10) *[Loop] Go to step 2.[8]*

C. Memetic Algorithm

MAS are population-based heuristic search schemes similar to genetic algorithms (GAS). GAS relies on the concept of biological evolution, but MAS, in contrast, mimic cultural evolution. While, in nature, genes are usually not modified during an individual's lifetime, memes are. They can be thought of as units of information that are replicated while people exchange ideas. A person usually modifies a meme before he or she passes it on to the next generation. As with genes, memes also evolve over time; good ideas may survive, while had ones may not [11].

The outline of the memetic algorithm : Algorithm MA:

```
Begin
Initialize population P;
For each individual i € P do i = Local-Search (i)
Repeat
For i = I to #crossovers Do
Select two parents $i_a$, $i_b$€ P randomly;
$i_c$ = crossover($i_a$,$i_b$) ;
$i_c$ = Local. Search($i_c$) ;
Add individual $i_c$ to P ;
End for;
For i = 1 to #mutations Do
Select an individual i € P randomly;
$I_m$, = Mutates (i);
$I_m$ = Local-Search($I_m$) ;
Add individual i, to P;
End for;
P = select (P);
If P converged then
For each individual i € P \ (best) Do
i = local Search (Mutates (i));
End if;
Until terminate = true;
End;
```

D. Edge Assembly Crossover (EAX)

A crossover operator, called edge assembly crossover (EAX). It was proposed by Nagata and Kobayashi in 1997. The EAX has two important features: preserving parents' edges using a novel approach and adding new edges by applying a greedy method, analogous to a minimal spanning tree. In this we select two individuals tours denoted as A and B, are selected as parents. EAX first merges A and B into a single graph denoted as R and then considered a powerful crossover operator. We called this even-cycle AB-cycle. All of the edges in this AB-cycle are then deleted from graph R. The same procedure is repeated until all of the edges in graph R are eliminated. Finally, in order to get a valid solution the EAX uses a greedy method to merge these distinct sub tours together. [11].

III. PROPOSED SOLUTION

The TSP is defined as a task of finding of the shortest Hamiltonian cycle or path in complete graph of N nodes. The TSP can be formally described as a graph G =(V,E), where V =(v_1, ..., v_n) is the set of vertices, E = (f(v_i , v_j): v_i, v_j belongs to V)is the set of edges.

A. Fitness function

The GAs is used for maximization problem. For the maximization problem the fitness function is same as the objective function. But, for minimization problem, one way of defining a 'fitness function' is as

F (x) =1/f(x)

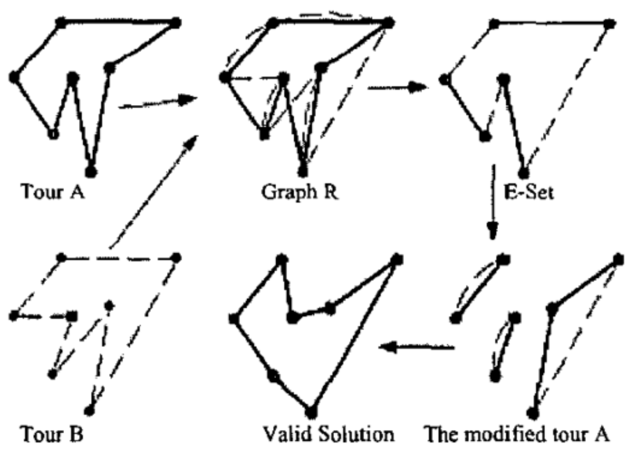


Figure 1. An Example of EAX[10]

Where *f*(x) is the objective function. Since, TSP is a minimization problem; we Consider this fitness function, where f(x) calculates cost (or value) of the tour represented by a Chromosome.

B. Algorithm

Let we have cities with their coordinates values. Now follow the steps

Step 1: Construct a tree or minimum spanning tree from the graph based on the group size. Root node is the starting point of the salesman.

Step 2: Construct the tours from each leaf node and from starting node in tree in the following way.

Step 3: Applying the GA to this tree.

I. Select any chromosome form the tree

II. Here we have two kind of crossover one is inside the chromosome and other in between two chromosomes.

a) One point - part of the first parent is copied and the rest is taken in the same order as in the second parent

b) Two point - two parts of the first parent are copied and the rest between is taken in the same order as in the second parent

c) None - no crossover, offspring is exact copy of parents

III. Mutation can be done to the chromosome in the following ways.

a) Normal random - a few cities are chosen and exchanged

b) Random, only improving - a few cities are randomly chosen and exchanged only if they improve solution (increase fitness)

c) Systematic, only improving - cities are systematically chosen and exchanged only if they improve solution (increase fitness)

d) None - no mutation

Step 4: Add the entire route from the entire computed tree (from step 3) with the route using EAX; consider two best route from the tree as A and B. If result of EAX is better than our A and B than take the result of EAX Otherwise take best route from A and B.

Step 5: Compute the fitness score from the route formed from step 5 using fitness function if new route has better fitness score than previous then replace the route.

Step 6: Repeat step 3 to step 5 until better fitness score is obtained or all computed tree (from Step 3) have been added to the route.

Step 7: return the best result.

IV. Simulation Result

There are 2 parameters to control the operation of the Genetic Algorithm:

1) Neighborhood / Group Size – Each generation, this number of tours are randomly chosen from the population. The best 2 tours are the parents. The worst 2 tours get replaced by the children. For group size, a high number will increase the likelihood that the really good tours will be selected as parents, but it will also cause many tours to never be used as parents. A large group size will cause the algorithm to run faster, but it might not find the best solution.

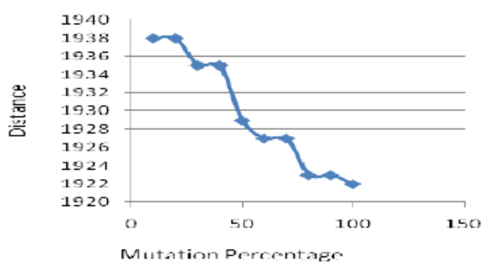
2) Mutation % - The percentage that each child after crossover will undergo mutation .When a tour is mutated, one of the cities is randomly moved from one point in the tour to another.

The starting parameter values are:

Parameter	Initial Value
Group Size	5
Mutation	3 %

In this simulation result, we have found the TSP route for 30 cities and changed the parameter like group size and mutation percentage. After analyses we observed that, in Fig. 1 with same group size if we increasing the mutation % then it slowly decreasing distance value of the tour and in fig. 2 with same mutation % if we increasing the group size then first it slowly decreasing the distance value of the tour but after a certain value of the group size, distance value of the tour will increase.

Fig. 2 Distance Vs Mutation Percentage



Sr. no	Mutation Percentage	Total Distance
1	10	1938
2	20	1938
3	30	1935
4	40	1935
5	50	1929
6	60	1927
7	70	1927
8	80	1923
9	90	1923
10	100	1922

Table 1

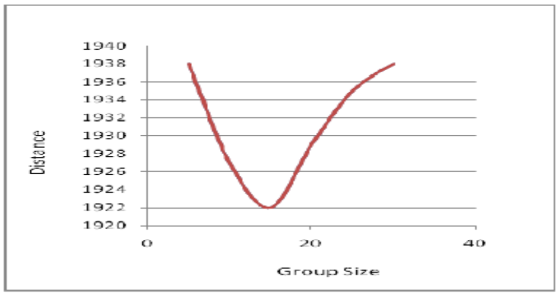


Fig.3 Distance Vs Group Size

Sr. no	Group Size	Total Distance
1	5	1938
2	10	1927
3	15	1922
4	20	1929
5	25	1935
6	30	1938

Table 2

V. COMPARISON

Comparison of TSPGA [11] with my proposed solution.
Cities
Location

Cities Names	Coordinates
A,B,C,D E,F,G,H I,J,K,L M,N,O	(110,54), (236,110), (153,151), (227,49), (307,176), (220,211), (341,90), (149,91), (335,40), (371,150), (218,161), (334,239), (148,227), (49,128), (183,39)

TSPGA Result [11].

Sequence	D_{min}
I-J-L-F-M-K-C-N-H-A-O-D-E-B-G	1298

Table III [11].

My proposed solution Result.

Sequence	D_{min}
A-H-O-D-I-G-J-L-E-B-K-F-M-C-H	1105

VI. CONCLUSION

In this paper "improving the solution of traveling salesman problem algorithm" based on NP-Hard problem. I have analyzed the result on different parameter like group size and mutation percentage. I observed that with same group size if we increasing the mutation percentage then it slowly decreasing distance value of the tour and with same mutation percentage if we increasing the group size then first it slowly decreasing the distance value of the tour but after a certain value of the group size, distance value of the tour will increase with the group size. I have also use EAX to improve the solution and compared my result with the exiting solution. In future work I am planning to reduce distance of the tour with further improvement in the algorithm.

REFERENCES

[1] Introduction to Algorithms, 2nd Ed. pp.1027-1033,2001.

[2] C.H. Papadimitriou and K. Stieglitz. "Combinatorial Optimization: Algorithms and Complexity". Prentice Hall of India Private Limited, India, 1997.

[3] X. P. Wang and L. M. Cao, Genetic Algorithm-Theory, Application and Software Realization. Xi'an, Shanxi: Xi'an Jiao Tong University Press, 2002.

[4] Zhu Qiang," A New Co-evolutionary Genetic Algorithm for Traveling Salesman Problem" ,IEEE ,2008.

[5] S. Chatterjee, C. Carrera, and L. A. Lynch, "Genetic Algorithms and Traveling Salesman Problems," European Journal of Operational Research, vol. 93, 1996.

[6] Budinich, M. (1996), A self-organizing neural network for the traveling salesman problem that is competitive with simulated annealing. Neural Computation, 8: 416-424.

[7] Traveling salesman problem, Computers & Operations Research, 30(5): 773-786.

[8] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. New York: Addison-Wesley Publishing Company, Inc., 1989.

[9] Y. Nagata, S. Yobayashi, "Edge assembly crossover: A highpower genetic algorithm for the traveling salesman problem," In Proceedings of the 7 International Conference on Genetic Algorithms (ICGA97), pp.450457, 1997.

[10] A Novel Memetic Algorithm with Random Multi-local-search: A case study of TSP Peng Zou', Zhi Zhou', Guoliang Chen', Xin Yao'.' 'National High Performance Computing Center at Hefei 230027 Hefei P.R.China 'Nature inspired computation and applications laboratory 'Dept. of Comp. Sci. and Tech., Univ. of Sci. and Tech. of China, 'School of Comp. Sci., The Univ. of Birmingham, Edgbaston, Birmingham B15 2TT, UK.

[11] Buthainah Fahran Al-Dulaimi, and Hamza A. Ali,"Enhanced Traveling Salesman Problem Solving by Genetic Algorithm Technique (TSPGA)" World Academy of Science, Engineering and Technology ,38, 2008.

AUTHORS PROFILE

Mohammad Junedul Haque is a lecturer at the College of Computers and Information Technology, Taif University, Saudi Arab. He holds master's degrees in computer science. His areas of interest are Algorithms,Data Mining, Image Processing and Networking.Mohammad Junedul Haque can be contacted by e-mail at junedulhaq@gmail.com.

Khalid Waheeb Magld is holding the position of vice dean for graduate studies and scientific research in Faculty of Computing and IT, King Abdulaziz University, Jeddah Saudi Arabia. He received first degree in Computer Science from Metropolitan State University in 1986 and Master in Computer Science in 1994 from University of Detroit, USA. He received his PhD from University of Bradford, United Kingdom. He had also been involved in many projects related to database design and data modeling with the IT center and other public and private sectors. His research interests include database design and data modeling, data mining, data retrieval, unicast and multicast in mobile adhoc networks, neural networks analysis and applications, web-based simulation, training and education and artificial intelligence.Khalid Waheeb Magld can be contacted by e-mail at kmagld@kau.edu.sa.