Efficiency Analysis of Genetic Algorithm and Ant Colony Optimization Techniques for Travelling Salesman Problem

Bajracharya Binod
Department of Software Engineering Technology and Management, School of Software Engineering
Jiading Campus, Tongji University
4800 Cao'An Highway Shanghai, P.R. China 201804
bbajracharya@qq.com; binodbajra@gmail.com

***Abstract*—In this paper, we present a genetic algorithm and discuss its methodologies to solve complex problem in combinatorial optimization problems e.g. travelling salesman problem. We analyze a real problem with the help of valid demonstrations and also compare this novel algorithm with very famous ant colony optimization techniques which is also used for solving the similar problem but using different approaches and parameters. The experimental results demonstrate the effectiveness of the proposed algorithm.**

***Keywords*—artificial intelligence; genetic algorithm; ant colony optimization; travelling salesman problem**

I. INTRODUCTION

The main idea of travelling salesman problem is that a salesman must visit "n" cities, passing through each city only once, beginning from one of the city that is considered as a base or starting city and returns to it. The cost of the transportation among the cities is given. The problem is to find the order of minimum cost route i.e., the order of visiting the cities in such a way that the cost is the minimum. It is an NP-hard problem in combinatorial optimization. It is important in operations research and theoretical computer science. The problem was first formulated in 1930 and is one of the most intensively studied problems in optimization.

A. *Purpose*

1) Objective

Our main objectives are to describe about a) novel algorithm, b) the tools and techniques that solve the travelling salesman problem, c) to investigate about new heuristic information based on artificial intelligent techniques so as to solve the travelling salesman problem for larger number of cities and finally, d) to analyze an efficiency of genetic algorithm and ant colony optimization techniques.

2) Current Techniques and Their Benefits and Limitations

Ant colony optimization technique has been proven a successful technique and applied to a number of combinatorial optimization problems and is taken as one of the high performance computing methods for travelling salesman problem. Ant colony optimization has very good search capability for optimization problems, but it still remains a computational bottleneck that this algorithm costs too much time to convergence and traps in local optima in order to find an optimal solution for travelling salesman problems.

II. MODEL AND ALGORITHM

Although, we can see a lot of variations in ant colony optimization algorithms, we have to research about key concepts of the new algorithm of artificial intelligence technique because of its easily solving complex problems and hence to improve the results of the search.

A. *Travelling Salesman Problem*

Reference [2]. succinctly summarizes the problem as, "Given a collection of cities, the travelling salesman must determine the shortest route that will enable him to visit each city precisely once and then return back to his starting point."

The number of possible routes is a permutation of the number of "n" cities to visit i.e. (n-1)!. The problem with permutation is that they grow in size remarkably quick. For e.g., the permutation of 10 is 3,628,800 but the permutation of 50 is gigantic, i.e. 3.041*10^64. This is an example of what mathematicians call NP-Complete Problems as in [2].

B. *Problem Description and Model*

- As more cities are added, the computational power required to solve the problem increases exponentially. An algorithm implemented on a computer that solves the travelling salesman problem for fifty cities would require an increase in computer power of a thousand-fold just to add an additional ten cities.
- Even with the today's modern computing power of computer, it is impractical or limited to solve such bigger problems which in turn gives arise to find a solution with alternate algorithm.
- Clearly, a "brute force" way becomes impossible for a large number of cities and alternate algorithm needs to

be employed if a solution is to be found for this problem.

III. A NEW SOLUTION

Thus, we have proposed genetic algorithm to overcome such problems and we have demonstrated this algorithm to solve real problems. The detail discussions about this algorithm are given in further section.

A. *Genetic Algorithm*

The algorithm consists of the following fundamental steps:

1) Fundamental Steps: These are the necessary steps of algorithm which are presented as follows.

a) Initialization: The chromosomes are randomly created. At this point, it is very important that the population is diverse. Otherwise, the algorithm may not produce good solutions.

b) Evaluation: Each chromosome is rated on how well the chromosome solves the problem at hand. A fitness value is assigned to each chromosome.

c) Selection: The fittest chromosomes are selected for propagation into the future generation based on how fit they are.

d) Recombination: Individual chromosomes and pairs of chromosomes are recombined, modified and then put back into the population.

2) Partially Matched Crossover Operators

The mechanisms used to produce valid “crossover operators” and “roulette wheel selection” of genetic algorithm are presented in this paper. All relevant terms from literature are used to adopt the standard naming conventions.

Each chromosome is encoded as a possible tour. For e.g., a tour of five cities might be encoded as 3,4,0,1,2. A difficulty with the travelling salesman problem is that a simple crossover will not work. Consider the following situation where crossover occurs at position 3.

TABLE I. CROSSOVER

Chromosome	*Encoded Tour*
Parent 1	1 2 3 4 5
Parent 2	3 5 2 1 4
Child 1	1 2 3 1 4
Child 2	3 5 2 4 5

A problem that has occurred here is that “child 1” has visited “city 1” two times, which is not allowed, and “child 2” has not visited “city 1” at all as shown in Table I. A defacement encoding must be used in order to make sure that only valid tours are produced. A well-known and perhaps the simplest to understand crossover operator is called “Partially Matched Crossover”. Reference [2]. explains this technique as follows:

TABLE II. CROSSING REGION PICKED UP

Chromosome	*Encoded Tour*
Parent 1	12x34x5
Parent 2	35x21x4

TABLE III. ESTABLISHED MAPPING

Mapping Table
3→2
4→1

A crossing region is picked as shown in Table II. Now, we cycle through each parent gene and swap the genes wherever a gene is found that occurs in the mapping as shown in Table III.

TABLE IV. FIRST ITERATION (MAPPING 3 WITH 2)

Chromosome	*Encoded Tour*
Child 1	1 3 2 4 5
Child 2	2 5 3 1 4

TABLE V. SECOND ITERATION (MAPPING 4 WITH 1)

Chromosome	*Encoded Tour*
Child 1	4 3 2 1 5
Child 2	2 5 3 4 1

The final crossed over genes are valid permutations with no duplicates as shown in Table V.

3) Roulette Wheel Selection

It is a technique of choosing members from the population of chromosomes in a way that is proportional to their fitness as in [2].

B. *Ant Colony Algorithm*

We know that this algorithm is inspired by observation of real ants. The colony of ants demonstrates complex behavior. The technique is carried out by ants laying down special chemicals called “pheromones”. As more ants use a particular trail, the pheromone concentration on it increases, hence attracting more ants. In our example, an artificial ant is placed randomly in each city and, during each iteration, chooses the next city to go to. This choice is governed by the “(1)”, as in [1].

Each ant located at city **i** hops to a city **j** selected among the cities that have not yet been visited according to the probability.

$$p_k(i, j) = \frac{\tau_{ij}^{\alpha} * d_{ij}^{\beta}}{\sum g \varepsilon J_k(i) \tau_{ig}^{\alpha} * d_{ig}^{\beta}} \qquad (1)$$

0, otherwise

where, "$p_k(i, j)$" is the probability that ant k in city **i** will go to city **j**. "g ε $J_k(i)$" is the set of cities that have not yet been visited by ant **k** in city **i**. "α" is the relative importance of the pheromone trail. "β" is the relative importance of the distance between cities.

Therefore, the probability that a city is chosen is a function of how close the city is and how much pheromone exists on that trail. Once a tour has been completed, pheromone evaporation on the edge is calculated. Then each ant deposits pheromone on the complete tour by a quantity which is calculated from the "(2)" as in [1].

$$\tau(i, j) = \rho.\tau(i, j) + {}^{m}\sum_{k=1} \Delta\tau_k(i, j) \qquad (2)$$

In "(2)", $\Delta\tau k = 1/L_k\ (i, j)\ \varepsilon$ tour (3)

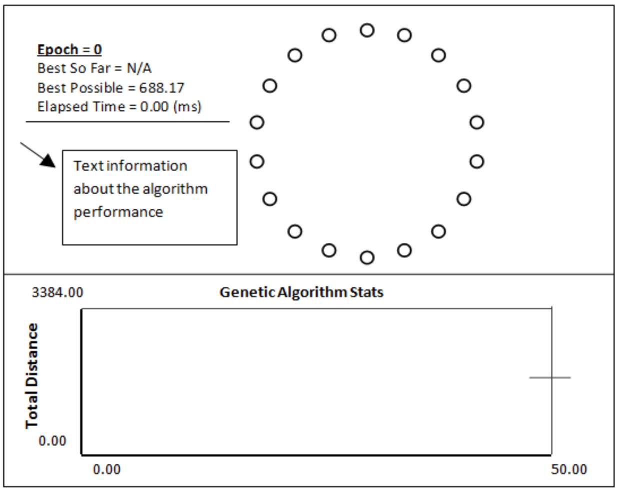
0, otherwise

where, "ρ.τ(i, j)" multiplies the pheromone concentration on the edge between cities **i** and **j** by **ρ,** which is called the "evaporation constant". This value can be set between 0 and 1. The pheromone evaporates more rapidly for lower values.

${}^{m}\sum_{k=1} \Delta\tau_k(i,j)$ is the amount of pheromone an ant **k** deposits on an edge, as defined by "L_k" which is the length of the tour created by this ant. Intuitively, short tours will result in higher levels of pheromone deposited on the edges.

IV. Experiment Results and Analysis

The genetic algorithm view is shown in the picture below.



Epoch = 0
Best So Far = N/A
Best Possible = 688.17
Elapsed Time = 0.00 (ms)

Text information about the algorithm performance

Fig. 1. A view of an application to find the shortest optimal route.

TABLE VI. RESULTS FOR DATASET 1: C40

Attributes	***Values***
Epoch	19
Best So Far	2196.51
Best Possible	757.79
Elapsed Time	27.65 ms

- "Epoch" represents the number of times the simulation has run through each cycle of the simulation.
- "Best So Far" is the shortest tour so far.
- "Best Possible" represents the optimal solution to the current tour.
- "Elapsed Time" is the computational time spent inside the algorithm.

The graph of the experiment as shown in Fig. 1 and Table VI is as shown below.

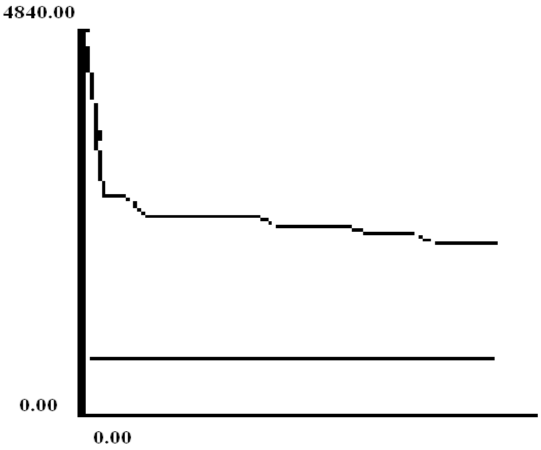


Fig. 2. The curved line is a plot of Epoch vs. shortest distance found by the genetic algorithm to date.

The line at the bottom of Fig. 2 represents the shortest possible tour. When the two lines converge, a solution has been found.

V. Performance and Efficiency Analysis

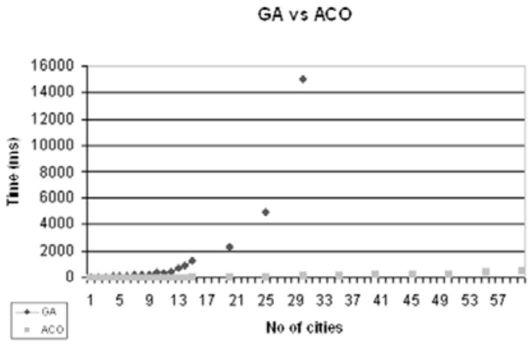


Fig. 3. Performance comparison between genetic algorithm and ant colony optimization.

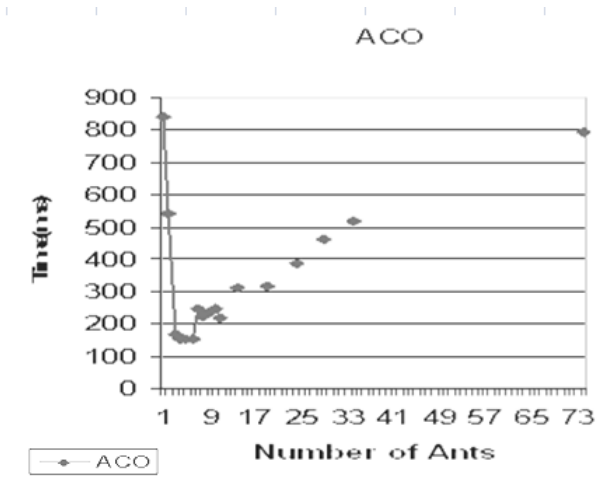


Fig. 4. Efficiency result after modification of certain parameters in ant colony optimization.

In Fig. 3, parameters used were city no: 40, α: 1, β: 1 and ρ: 0.6. In Fig. 4, result is shown after modification of heuristics parameters of ant colony optimization techniques. Again, based on those results, analysis of overall performance and efficiency were carried out to decide whether or not they have yielded minimized time to solve the problems.

VI. Conclusions

In this proposed novel algorithm, we have showed how real problems of travelling salesman could be easily solved with genetic algorithm. We were able to find a route just over thrice as good as our original and probably pretty close to optimum. We have tried to compare the performance that we get with genetic and ant colony algorithms. Hence, the results obtained show that genetic algorithm can solve complex issues efficiently and is able to find the shortest optimal route in a feasible manner. For finding tour solution of TSP benchmark problems by our algorithm, it might be possible to add different routines or procedures that would optimize the genetic algorithm parameters and obtain more feasible solutions.

Acknowledgement

The author would like to thank all the team including Prof. Cao Buyang and Aro Lee for their efforts and valuable suggestions. Most importantly, none of this would have been possible without the love and patience of my family throughout my Postgraduate Masters research and thesis. My heartfelt thanks also extend to all my colleagues and friends for their help, support, interest and valuable hints for discussions about research.

References

[1] M. Dorigo, and L.M. Gambardella, "Ant colonies for the travelling salesman problem," BioSystems, vol. 43, issue 2, pp. 73-81, Elsevier, 1997. (*references*)

[2] M. Buckland, "AI techniques for game programming," Premier Press: USA, 2002, pp.118-132. (*references*)

[3] J.H. Holland, "Adaptation in natural and artificial systems," Ann Arbor: University of Michigan Press, 1929, (OCoLC) 595057694.

[4] M. Jones, "AI application programming," David Pallali.

[5] T.D. Gwiazda, "Genetic algorithms reference vol.1 crossover for single objective numerical optimization problems," Lomianki., 2006.

[6] Ahmed and H. Zakir, "Genetic algorithm for the travelling salesman problem using sequential constructive crossover operator," International Journal of Biometric and Bioinformatics 3.6, 2010.

[7] S.N. sivanandam and S.N. Deepa, "Introduction to genetic algorithms," Springer Science & Business Media, Decemeber 2007.

[8] Y. Xuesong, Z. Zhang, W. Luo, W. Li, W. Chen and H. Liu, "Solve travelling salesman problem using particle swarm optimization algorithm," vol. 9, issue 6, no. 2, pp. 264, International Journal of Computer Science Issues (IJSCI), Nov 2012.

[9] Dalip and N. Kaur, "An efficient hybrid genetic algorithm for performance enhancement in solving travelling salesman problem," vol. 3, issue 11, pp.3502, International Journal on Computer Science and Engineering, November 2011.

[10] D.E. Goldberg, "Genetic algorithms in search, optimization and machine learning", Addison-Wesley Longman Publishing Co., Inc., New York, NY, USA, 1989.