

A Comparative Study of GA and SA for Solving Travelling Salesman Problem

Pradeep Vaishnav

M.Tech(Pursuing), Cad/Cam-Robotics,
Department of Mechanical Engineering
Christian College of Engineering and Technology
Kailash Nagar, Bhilai, India

Rahul Patil

Assistant Professor,
Department of Mechanical Engineering
Christian College of Engineering and Technology
Kailash Nagar, Bhilai, India

Abstract—This study discusses the travelling salesman problem and by presenting the history of the methods used in solving the travelling salesman problem intends to solve the problem by Meta-heuristic Algorithms such as Simulated Annealing and Genetic Algorithm. In this paper, Simulated Annealing and Genetic Algorithm are presented for Random Travelling Salesman Problem (RTSP). Random Travelling Salesman Problem is a variant of TSP. All TSP dataset used here are constructed randomly and then GA and SA model are applied on those data sets. Simulated Algorithm has higher efficiency in solving Traveling Salesman Problem than Genetic Algorithm. Experimental results are discussed based on several criteria like time, quality and accuracy.

Keywords- Random Travelling salesman Problem(RTSP), Simulated Annealing(SA), Genertic Algorithm(GA)

I. INTRODUCTION

The travelling salesman problem(TSP) is the problem of finding a shortest closed tour which visits all the cities in a given set. Real ants are capable of finding the shortest path from a food source to the nest without using visual cues [1]. They are deposit pheromones according to the quality of the path they find, this allow capable of adapting to changes in the environment, for example finding a new shortest path once the old one is no longer feasible due to a new obstacle consider [1]. In this paper two widely used nature inspired heuristic approaches Simulated Annealing and Genetic Algorithm to solve tsp are considered. In this research Sa and GA are compared for TSP instances. Given a set of cities and known distances between each pair of cities, the TSP is the problem of finding a tour that visits each city exactly once and that minimizes the total distance travelled[2].

The paper starts with a discussion on Random Travelling Salesman Problem in section 2. A discussion on GA model is presented and discusses the proposed GA model in section 4. A discussion and implementation details of SA model is presented in section 3. Implementation and results of both algorithms compared in section 5. Finally the paper ends with conclusion in section 6.

II. RANDOM TRAVELLING SALESMAN PROBLEM

In the *Travelling Salesman Problem* (TSP) a no. of cities are provided with the objective of finding a minimum length closed tour that visits each city once returning to the starting city. There are various types of TSP. A *symmetric* TSP (STSP) is one where the distance between any two cities is same from either side. In an *asymmetric* TSP (ATSP) the distance is not same. Random TSP is also one of the variants of TSP. In this

TSP, I used randomly generated TSP instances on TSPLIB. All the data sets are created with randomly generated city coordinates. I have applied a computational model inspired by Genetic Algorithm and Simulated Algorithm.

III. THE SIMULATED ANNEALING (SA) MODEL

This section first describes the concepts of Simulation Annealing method. How a process of cooling a molten metal slowly with some temperature setting is simulated in an algorithm, which can be used to solve various optimization problems.

A. Simulated Annealing (SA)

Simulated Annealing is a generic probabilistic meta-algorithm used to find an approximate solution to global optimization problems. It is inspired by annealing in metallurgy which is a technique of controlled cooling of material to reduce defects. The Simulated Annealing algorithm starts with a random solution [11]. A random nearby solution is formed by every iteration. If this solution is a better solution, it will replace the current solution.

B. SA Model for Random TSP

This section will focus on applying Simulated Annealing (SA) model to the Random Travelling Salesman Problem (TSP). In this model, a random number generator is used to determine an initial solution. Since we are dealing with a TSP with n cities, it starts at city 1, the solution is an ordered list of n integers. This list starts with the number 1 and contains the numbers 1 through n exactly once. For example, in a 6-city problem, the array (1,4,3,5,6,2) means that the route starts at city 1 and then travels to cities 4, 3, 5, 6, and 2, in that order, before returning to city 1. As stated in the general discussion, Simulated Annealing is a computational methodology, not a particular algorithm. You have an algorithm, and all you need is the seed for the random number generator to run the simulation. A Monte Carlo step is determined by randomly choosing two *positions* in the array (except the first) and switching the cities in these positions.

For the procedure chosen here, a proportional reduction of the temperature yields a better final result than a constant reduction in every case but one. In addition, the proportional reduction appears to converge better since there is a smaller change in the final length when the seed to the random number generator is changed. The constraints that affect the outcome of the algorithm are the initial temperature decreases and the stopping condition of the algorithm. By adjusting these values. I run the SA and to see that how algorithm responds.

Pseudo-code of SA model for Random TSP is given below.

1. Generate a random dataset or add a standard dataset from TSPLIB (as a part of initializing the population);
2. While (termination criteria not met)
3. Generate new solutions;
4. Access new solutions;
5. If new solution accepted;
6. Update storage;
7. Adjust temperature;
8. Evaluate and update solutions;

Figure 1. Pseudo-Code Of SA-RTSP Model

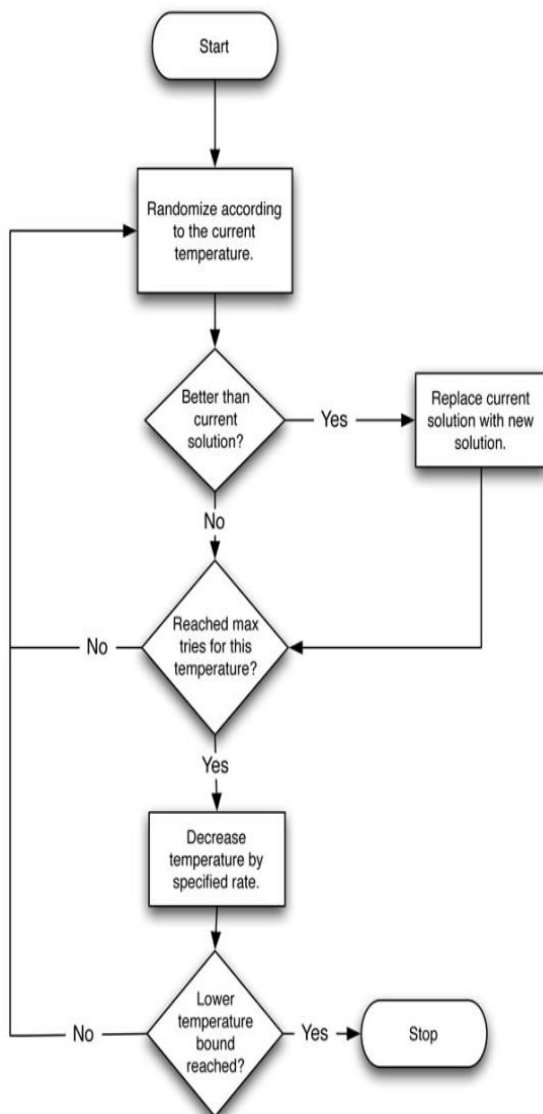


Figure 2. Flow Chart Of Sa-Rtsp Model

IV. GENETIC ALGORITHM (GA) MODEL

Genetic Algorithms (GAs) are a family of computational models inspired by evolution. The study of Darwin on evolution and natural selection has inspired the development of this computational model for optimization [19]. GAs are evolutionary techniques based on the application of selection, mutation, and recombination to a population of solutions. GAs

have been successfully applied to a variety of optimization problems, including TSP, Job shop Scheduling, Vehicle routing problems. GAs are directed in their searching nature. This algorithm can generate global optimum if it is given a well defined problem with enough time. This makes them well suited to a problem like Traveling Salesman Problem.

In the field of artificial intelligence genetic algorithm is a search heuristic that mimics the process of natural evolution. Genetic Algorithm belongs to class of evolutionary algorithm. GA begin with various problem solution which are encoded into population, a fitness function is applied for evaluating the fitness of each individual, after that a new generation is created through the process of selection, crossover and mutation. After the termination of genetic algorithm, an optimal solution is obtained. If the termination condition is not satisfied then algorithm continues with new population.

Pseudo-code of SA model for Random TSP is given below.

1. Generate a random dataset or add a standard dataset from TSPLIB;
2. Initialize the population of genes;
3. Optimize the population.(apply local search)
4. Evaluate the population;
5. While (termination criteria not met)
6. Apply Selection;
7. Apply Crossover;
8. Apply Mutation;
9. Optimize population;
10. Evaluate and Update population;

Figure 3. Pseudo-Code Of GA-RTSP Model

The GA ends when a predefined number of evolutions are reached or all chromosomes converge to the same mapping. This genetic algorithm randomly chooses chromosomes. Crossover is the process of swapping certain sub-sequences in the selected chromosomes. Mutation is the process of replacing certain sub-sequences with some mapping choices new to the current population. Both crossover and mutation are done randomly. After crossover and mutation, a new population is generated. Then this new population is evaluated, and the process starts over again until some stopping criteria are met.

The stopping criteria can be, for example,

- 1) No improvement in recent evolutions
- 2) Number of Iterations
- 3) A cost bound

The procedure of GA search for Random TSP is as follows:

- 1) Population generation: A population is a set of tours and each represents a possible solution. The initial population is generated randomly. Every candidate of the tour is generated randomly and put in the population.
- 2) Population evaluation: Each tour (chromosome) is assigned a fitness value. The fitness of population is evaluated based on the objective function.
- 3) Crossover operation: Crossover operation chooses a pair of tours (chromosomes) and mats them to produce a new offspring. One point crossover mechanism is used here.

4) Mutation: Mutation randomly chooses a chromosome, then randomly selects a city within the chromosome, and randomly reassigns it to a new tour.

The flow chart of the Genetic Algorithm Model is given below

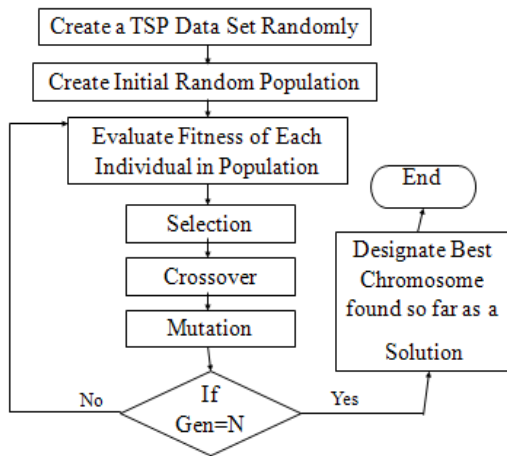


Figure 4. Flow Chart Of GA-RTSP Model

V. IMPLEMENTATION AND RESULT

Here I implemented the simulated annealing and genetic algorithms to apply it to randomly generated TSP instances. The algorithms are coded in MATLAB 7.0 and executed on windows XP with Pentium dual core CPU and 1 GB RAM. I have generated TSP instances within the range of 10 to 200 cities. The length of tour for each city solved in MATLAB for each algorithm. The experimental results are presented in table-1.

TABLE I EXPERIMENTAL RESULTS

City Problem	Length of Tour	
	SA Model	GA Model
10	5.5839	28.3030
20	9.1664	40.2914
30	11.1057	43.1653
40	16.1983	48.4221
50	22.1901	60.0268
60	27.3245	64.9082
75	33.6674	72.2302
95	50.5301	78.0420
100	51.5866	82.2717
105	52.6431	86.5015

VI. CONCLUSION

A model of SA-RTSP and GA-RTSP has been introduced to solve Random TSP. The model has been tested on a set of randomly generated TSP problems. This paper presents a comparative view of most widely used optimization algorithm techniques namely SA and GA. Section 5 shows the

experimental results obtained on TSP problems. It shows that SA provides more better solution compared to GA.

REFERENCES

- [1] M. Dorigo, L. Gambardella, "Ant colonies for the Traveling salesman problem." Biosystems 43, pp. 73-81, 1997.
- [2] Jinhui Yang, Xiaohu Shi, Maurizio Marchese, Yanchun Liang. "An ant colony optimization method for generalized TSP problem". Prog Nat Sci 18 (2008) 1417-1422.
- [3] Shivali Puri and Dr. Baljit Singh Khchra, "A Comparative study of GA and ACO for solving travelling salesman problem", ICRTEDC Vol.1, Spl. Issue 2 (May 2014), ISSN:1694-2345
- [4] Xuesong Yan, Can Zhang et al. "Solve travelling salesman problem using particle swarm optimization algorithm" International Journal of computer science issues, Vol.9, issue 6, No.2, november 2012, ISSN:1694-0814
- [5] Seyed Ahmad Sheibat Alhamdy al. "Solving travelling salesman problem using Ant colony optimization algorithm and comparing with Tabu search, Simulated annealing and genetic algorithm" Journal of Applied Science Research, 8(1):434-440, 2012
- [6] Bharat V. Chawda and Nitesh M. Sureja, "An ACO Approach to solve a variant of tsp" IJAR CET Vol.1, Issue 5, July 2012. ISSN:2278-1323
- [7] Sapna Katiyar et al. "Implementation of travelling salesman problem using ant colony optimization" IJERA, Vol.4, June 2014, pp.63-67
- [8] Saloni Gupta and Poonam Panwar, "Solving travelling salesman problem using Genetic Algorithm" IJARCSSE, Vol.3, Issue 6, June 2013, ISSN:2277-128X
- [9] Preet Walia et al. "Comparison on the Performance of Genetic Algorithm and Ant Colony Optimization for Solving the Travelling Salesman Problem", IJARCSSE, Volume 4, Issue 2, February 2014 ISSN: 2277 128X
- [10] Sharad N. Kumbharana and Prof. Gopal M. Pandey, "A Comparative Study of ACO, GA and SA for Solving Travelling Salesman Problem", International Journal of Societal Applications of Computer Science, Vol 2 Issue 2 February 2013, ISSN 2319 - 8443
- [11] Nitesh M. Sureja and Bharat V. Chawda, "Random Travelling Salesman Problem using SA", International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 4, April 2012)
- [12] Sudhanshu Prakash Tiwari et al. "A Survey of Metaheuristic Algorithms for Travelling Salesman Problem" International Journal Of Engineering Research & Management Technology, September- 2014 Volume 1, Issue-5 Page 72.
- [13] Km. Shweta and Alka Singh, "An Effect and Analysis of Parameter on Ant Colony Optimization for Solving Travelling Salesman Problem", IJCSMC, Vol. 2, Issue. 11, November 2013, pg.222 - 229
- [14] Neha and Baijnath Kaushik "Improved Ant Colony System for Solving TSP", IJITKM, Volume 6, No. 2, July-December 2013 pp. 165-169.
- [15] Kanar Shukr Mohammed, "Modified Ant Colony Optimization for Solving Traveling Salesman Problem", International Journal of Engineering & Computer Science, Vol:13, No:05, October 2013
- [16] Zainudin Zuhri and Irving Vitra Papatungan, "A Hybrid Optimization Algorithm Based on GeNETIC Algorithm and Ant Colony Optimization", International Journal of Artificial Intelligence & Applications (IJAI), Vol. 4, No. 5, September 2013
- [17] Krishna H. Hingrajiya et al., "An Ant Colony Optimization Algorithm for Solving Travelling Salesman Problem" International Journal of Scientific and Research Publications, Volume 2, Issue 8, August 2012
- [18] R. N. Mondal, M. R. Hossain and S. K. Saha, "An Approach for Solving Traveling Salesman Problem" International Journal of Applied Operational Research, Vol. 3, No. 2, pp. 15-26, Spring 2013.
- [19] Nitesh M. Sureja and Bharat V. Chawda, "Random Travelling Salesman Problem using GA", International Journal of Computing, Volume 2, Issue 2, April 2012.