

Solving TSP Using Various Meta-Heuristic Algorithms

<http://dx.doi.org/10.3991/ijes.v1i2.3233>

Divya Gupta

Maharaja Surajmal Institute Of Technology, New Delhi, India

Abstract—Real world problems like Travelling Salesman Problem (TSP) belong to NP-hard optimization problems which are difficult to solve using classical mathematical methods. Therefore, many alternate solutions have been developed to find the optimal solution in shortest possible time. Nature-inspired algorithms are one of the proposed solutions which are successful in finding the solutions that are very near to the optimal. In this paper, Classical TSP (CTSP) along with its variant Random TSP (RTSP) are solved using various meta-heuristic algorithms and their performance is compared on the basis of tour length. Results show that the Nature-inspired algorithms outperform both Traditional and Evolutionary algorithms and obtain optimal solutions.

Index Terms—Evolutionary Algorithms, Nature-Inspired Algorithms, Simulated Annealing, Travelling Salesman Problem

I. INTRODUCTION

The Travelling Salesman Problem (TSP) [1] is a well-known NP-hard Optimization problem whose goal is to find the shortest route possible for a salesperson to take in visiting N cities. This type of problem appears in many forms with some engineering applications that include Vehicle routing [2], scheduling problems [3], integrated circuit designs [4], physical mapping problems [5], and constructing phylo-genetic trees [6].

A large number of methods have been developed for solving TSPs. Few of them are Simulated Annealing [7], Tabu-Search Algorithm [8], Genetic Algorithm [9], Memetic Algorithm [10], Ant Colony Optimization Algorithm [11], Bee Colony Optimization Algorithm [12], Firefly Algorithm [13], Cuckoo Search Algorithm [14] and many more. Although Evolutionary Algorithms (EA) such as Genetic Algorithm (GA) and Memetic Algorithm (MA) are appropriate for solving difficult optimization problems, for which the traditional algorithms such as Simulated Annealing (SA) and Tabu-Search (TS) algorithms are less efficient, the EAs remain inefficient to find the optimal solution to TSPs.

Many approaches such as designing TSP-specific operators, incorporating local searches, and maintaining population diversity are considered to be promising ways to solve TSPs. These methods have been incorporated in Nature-Inspired algorithms such as Ant Colony Optimization Algorithm (ACO), Bee Colony Optimization Algorithm (BCO), Firefly Algorithm (FA) and Cuckoo Search Algorithm (CS) which efficiently prevent premature convergence of TSP.

This paper examines two classical, two evolutionary and four nature-inspired algorithms based on their

qualities of their solutions and mechanisms by which edges that appear in a known optimal tour are preserved and added to obtain the minimized and optimized tour length.

Rest of the paper is organized as follows: Section II gives a detailed description of the Travelling Salesman Problem and its variant; Section III briefly discusses all eight algorithms used in this work; experimental results are shown in Section IV and finally in Section V the work is concluded.

II. TRAVELLING SALESMAN PROBLEM

The Travelling Salesman Problem (TSP) represents a large class of problems known as combinatorial optimization problems. These are difficult to solve using traditional methods and if solvable, computations tend to be very time consuming. Hence we settle for approximated results which in less time, end up giving a result that isn't necessarily the best tour, but instead a tour that is close to the best tour.

A form of the TSP was introduced by Euler in 1759 and later in 1948 Rand Corporation formally named and introduced it [15]. It defined the Classical Travelling Salesman Problem (CTSP) as a problem where starting from one city it is required by the salesman to visit every other city only once in a way that the total distance covered is minimized. As the number of cities increases, the complexity of the problem increases exponentially due to the number of possible solutions increased very heavily.

In this paper, CTSP and a variant of TSP named as Random Travelling Salesman Problem (RTSP) are solved using conventional, evolutionary and bio-inspired algorithms. The performances are compared on the basis of tour length. For CTSP, six available datasets on TSPLIB [16] are used and for RTSP, four datasets are randomly generated having all the city coordinates generated in a pre-defined range from 0 to 100.

III. META-HEURISTIC ALGORITHMS

Two classical (Tabu-search and Simulated Annealing), two evolutionary (Genetic and Memetic), and four nature-inspired algorithms (Ant Colony Optimization, Bee Colony Optimization, Firefly and Cuckoo-Search) have been used for comparison in this work. These are briefly described along with their pseudo codes in following section.

A. Simulated Annealing (SA)

Simulated Annealing (SA) is the oldest probabilistic meta-heuristic algorithm and one of the first algorithms having ability to avoid being trapped in local minima. It is inspired by the process of annealing in metallurgy. In this process a material is heated and slowly cooled into solid crystal state with minimum energy and larger crystal size to reduce defects in metallic structures. Once the system has cooled, the configuration will represent a sample at or close to a global optimum. Pseudo code of SA [17] for solving TSP is given as follows:

Pseudo Code for SA-TSP

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;

B. Tabu-Search Algorithm (TS)

Tabu search is an optimization technique for solving permutation problems. In this technique, we start with an arbitrary permutation and make a succession of moves to transform this permutation into an optimal one (or as close to the optimum as possible). In this particular setting, it is equivalent to starting with a randomly generated tour and making a succession of edge swaps trying to reduce the cost of the tour until we can find the minimum cost. Pseudo code of TS [18] for solving TSP is given as follows:

Pseudo Code for TS-TSP

1. Generate a random dataset or add a standard dataset from TSPLIB;
2. Initialize population and clear the tabu list;
3. While (termination criteria not met)
4. Generate neighbours of the current seed solution by a neighbourhood structure
5. If aspiration criteria is not satisfied
6. Select the “best” neighbour which is not tabu is as new seed
7. Update the tabu list
8. Else Store the aspiration solution as the new seed and the best solution
9. Output Optimization result

C. Genetic Algorithm (GA)

A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm repeatedly modifies a

population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution. Pseudo code of GA [19] for solving TSP is given as follows:

Pseudo Code for GA-TSP

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;

D. Memetic Algorithm (MA)

A Memetic Algorithm is a search technique inspired by Darwin’s principles of natural evolution [19] and Dawkin’s notion of a meme [20]. Meme is a unit of cultural evolution capable of individual learning. Every meme gains some experience through a local search before going in to evolution of new generations. Memetic Algorithm is also known as hybrid GA as it uses techniques inspired by natural selection such as mutation, selection, and crossover with the addition of local search. These techniques use same parameters as in GA added with the parameters of local search. Pseudo code of MA [21] for solving TSP is given as follows:

Pseudo Code for MA-TSP

1. Generate a random dataset or add a standard dataset from TSPLIB;
2. Initialize the population of memes;
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;

E. Ant Colony Optimization Algorithm (ACO)

ACO algorithm mimics the behavior of real life ants and on how they interact with each other. The basic philosophy of the algorithm involves the movement of a colony of ants through the different states of the problem influenced by two local decision policies, viz., trails and attractiveness and two mechanisms, viz., trail evaporation and daemon actions. The algorithm aims to search for an optimal path based on the behavior of ants seeking a path

between their colony and a source of food. Thereby, each such ant incrementally constructs a solution to the problem. Pseudo code of ACO [22] for solving TSP is given as follows:

Pseudo Code for ACO-TSP

1. Generate a random dataset or add a standard dataset from TSPLIB;
2. Initialize the pheromone trails;
3. While (termination criteria not met)
4. Construct Solutions;
5. Apply Local Search;
6. Update Trails;
7. Evaluate the best solutions;

F. Bee Colony Optimization Algorithm (BCO)

BCO algorithm is based on the food foraging behavior of swarms of honey bees. Honey bee colonies have a decentralized system to collect food and can adjust the searching pattern precisely in order to enhance the collection of nectar. The basic idea concerning the algorithm is that foraging bees have a potential solution to an optimization problem in their memory and this potential solution corresponds to the location of a food source and has an aggregated quality measure (i.e., value of the objective function). Pseudo code of BCO [23] for solving TSP is given as follows:

Pseudo Code for BCO-TSP

1. Generate a random dataset or add a standard dataset from TSPLIB;
2. Initialize the population;
3. While (termination criteria not met)
4. While (all bees have not built a complete path)
5. Observe Dance;
6. Forage By TransRule;
7. Perform Waggle Dance;
8. Optimize population;
9. Evaluate and Update population;

G. Firefly Algorithm (FA)

FA is a swarm-based algorithm based on the flashing behavior of fireflies. Fireflies or lightning bugs belong to a family of insects that are capable to produce natural light to attract a mate or prey. In firefly algorithm, a firefly population is placed in random locations in the search space where the fireflies represent a candidate solution. The flashing light of the fireflies is formulated in such a way that it gets associated with the objective function to be optimized. Pseudo code of FA [24] for solving TSP is given as follows:

Pseudo Code for FA-TSP

1. Generate a random dataset or add a standard dataset from TSPLIB (as a part of initializing population of fireflies);
2. While (termination criteria is not met)

3. Find the most attractive firefly;
4. Move other fireflies towards it if in range;
5. Else move fireflies randomly;
6. Select the brightest fireflies representing the solution;

H. Cuckoo- Search Algorithm (CS)

CS is a population based optimization algorithm influenced by the parasitic behavior of the cuckoo bird whose female lays their eggs in the nests of other birds. In CS algorithm, a set of nests with one egg each are placed in random locations in the search space where the eggs represent a candidate solution. A search pattern called Levy Flights is used which is considered more efficient than random walks or Brownian motions. Pseudo code of CS [25] for solving TSP is given as follows:

Pseudo Code for CS-TSP

1. Generate a random dataset or add a standard dataset from TSPLIB (as a part of initializing population of cuckoo nests);
2. While (termination criteria is not met)
3. Get a cuckoo randomly (say, i) and replace its solution by performing Lévy flights;
4. Evaluate its quality/fitness;
5. Choose a nest among n (say, j) randomly;
6. Replace j by the new solution if fitness value of i is more than fitness of j;
7. Abandon worse nests build new ones;
8. Keep the best solutions/nests;
9. Select the best nests representing the solution;

IV. EXPERIMENTAL RESULTS

All the eight meta-heuristic algorithms are implemented with some modifications in few parameters in order to adapt it to solve travelling Salesman problem. The experimental environment is implemented in MATLAB programs and executed on a DELL Studio15 Computer with the configuration of Intel Core I3 CPU M370 at 2.40 GHz and 4GB RAM. TSP datasets are standard TSP instances downloaded from TSPLIB for solving CTSP. For RTSP datasets are randomly generated. The algorithms run till the termination criteria are satisfied. More than one termination criteria in the model such as predefined number of iterations, stagnations in the result, time-limit etc. can be used but here stagnation in the result is used as the termination criteria. Results obtained are shown in table 1 and table 2 and are averaged over 25 runs of all the models for each data set.

Table 1 and Table 2 show performance comparison of all the eight meta-heuristic algorithms for CTSP and RTSP respectively. For both the problems Cuckoo Search Algorithm emerged as a winner with best solutions. Also the bio-inspired algorithms performed much better than the other two traditional and evolutionary algorithms.

V. CONCLUSION

Searching an optimal route for a TSP is very important to save time and cost. In this work, two classical, two evolutionary and four bio-inspired algorithms were implemented to find the best route for a classical and a random TSP. It was observed that the bio-inspired algorithms provide feasible methods for TSPs and can attain better performance than other methods. Results obtained also showed that the Cuckoo Search algorithm produces the best solutions in terms of tour length. It was also observed that further improvement in the bio-inspired algorithms could be done by fine-tuning their self-adaptive parameters and by using hybrid coding.

ACKNOWLEDGMENT

The author is thankful to Director, MSIT for his constant support and motivation for continuing the research work.

REFERENCES

- [1] Laporte, G. "The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms", *European Journal of Operational Research*, 59, 231-247., 1992. [http://dx.doi.org/10.1016/0377-2217\(92\)90138-Y](http://dx.doi.org/10.1016/0377-2217(92)90138-Y)
- [2] G. Clarke and J.W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Oper. Res.*, vol. 12, pp. 568-581, 1964.G. <http://dx.doi.org/10.1287/opre.12.4.568>
- [3] D. Whitley, T. Starkweather, and F. D'Ann, "Scheduling problems and traveling salesman: The genetic edge recombination operator," in *Proc.3rd Int. Conf. Genetic Algorithms*, 1989, pp. 133-140.
- [4] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 498-516, 1983. <http://dx.doi.org/10.1126/science.220.4598.671>
- [5] F. Alizadeh, R. M. Karp, L. A. Newberg, and D. K. Weisser, "Physical mapping of chromosomes: A combinatorial problem in molecular biology," in *Proc. 4th ACM-SIAM Symp. Discrete Algorithms (SODA)*, 1993, pp. 52-76.
- [6] C. Korostensky and G. H. Gonnet, "Using traveling salesman problem algorithms for evolutionary tree construction," *Bioinformatics*, vol. 16, no. 7, pp. 619-627, 2000. <http://dx.doi.org/10.1093/bioinformatics/16.7.619>
- [7] David Bookstaber, "Simulated Annealing for Traveling Salesman Problem", Spring, 1997
- [8] Glover F., Tilliard E., Werra D.E., "A user's guide to Tabu Search," *Annals of Operations Research*, 1993, 41, 3-28. <http://dx.doi.org/10.1007/BF02078647>
- [9] Holland, J.H. "Adaptation in Natural and Artificial Systems". MIT Press, 1992.
- [10] P. A. Moscato, "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Toward Memetic Algorithms," Caltech, Pasadena, CA, Tech. Rep. California Technical Institute concurrent computation program report 826, 1989.
- [11] M. Dorigo, L. Gambardella, "Ant colonies for the Traveling salesman problem." *Biosystems* 43 (1997): (73-81). [http://dx.doi.org/10.1016/S0303-2647\(97\)01708-5](http://dx.doi.org/10.1016/S0303-2647(97)01708-5)
- [12] K. v. Frisch, "Decoding the language of the bee," *Science*, vol. 185, no. 4152, pp. 663 - 668, 1974. <http://dx.doi.org/10.1126/science.185.4152.663>
- [13] Yang, X.S. "Firefly Algorithms for Multimodal Optimization". In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009. LNCS, vol. 5792, pp. 169-178. Springer, Heidelberg, 2009
- [14] Yang, X.S.: Nature-inspired Metaheuristic Algorithm. Luniver Press, 2nd Edition (2010) .
- [15] Michalewicz, Z. "Genetic Algorithms + Data Structures = Evolution Programs", New York, Springer-Verlag, 1992. <http://dx.doi.org/10.1007/978-3-662-02830-8>
- [16] Reinelt, G.: TSPLIB 95. Technical report, Research Report, Institut für Angewandte Mathematik, Universität Heidelberg, Heidelberg, Germany (1995).
- [17] Nitesh M Sureja, Bharat V Chawda, "Random Travelling Salesman Problem using SA", *International Journal of*
- [18] *Emerging Technology and Advanced Engineering*, Volume 2, Issue 3, March 2012.
- [19] Battiti R., Tecchiolli G., 1994, The Reactive Tabu Search, *ORSA Journal on Computing*, 6 (2), 126-140. <http://dx.doi.org/10.1287/ijoc.6.2.126>
- [20] Nitesh M Sureja, Bharat V Chawda, "Random Travelling Salesman Problem using Genetic Algorithms", *IFRSA's International Journal Of Computing*, Volume 2, Issue 2, April 2012.
- [21] Nitesh M. Sureja, Bharat V. Chawda, "Memetic Algorithm a Metaheuristic Approach to Solve RTSP", *IJCSEITR*, ISSN 2249-6831, Vol. 3, Issue 2, pp. 183-186, June 2013
- [22] Nitesh M Sureja, Bharat V Chawda, "An ACO Approach to Solve a Variant of TSP", *International Journal of Advanced Research in Computer Engineering & Technology*, Volume 1, Issue 5, July 2012.
- [23] Sunil S. Shah, Prashant B. Swadas, Bharat V. Chawda, "Travelling Salesman Problem Solutions using Various Heuristic Algorithms", *ICIKR-ETS-2012*, Rajkot, March, 2012
- [24] Gilang Kusuma Jati and Suyanto, "Evolutionary discrete Firefly Algorithm for Travelling Salesman Problem", *ICAIS*, pp. 393-403, springer-Verlag Berlin Heidelberg, 2011.
- [25] Aziz Ouaraab, Belaïd Ahiod, Xin-She Yang "Discrete Cuckoo Search Algorithm for Travelling Salesman Problem", *Neural Computing and Applications*, springer-Verlag Berlin Heidelberg, April 2013. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

AUTHORS

Divya Gupta is with the Department of Computer Science, Maharaja Surajmal Institute of Technology, Guru Gobind Singh Indraprastha University, New Delhi, India (e-mail: omerhub.divya@gmail.com).

Submitted 05 October 2013. Published as re-submitted by the author 02 November 2013.

TABLE I: PERFORMANCE COMPARISON FOR CTSP

TSP File	No of Cities	SA	TS	GA	MA	BCO	ACO	FA	CS
City10	10	64.19	64.20	60.60	60.55	60.87	60.50	60.50	60.50
Ulysses 16	16	8977	8979	7399	7808	7567	7400	7398	7235
City29	29	19180.81	23513	9074.12	9106.73	20703	9999.29	9072.98	9065.98

PAPER
SOLVING TSP USING VARIOUS META-HEURISTIC ALGORITHMS

Oliver30	30	423.74	424.99	423.74	437.66	431.12	452.96	423.50	421.56
Eil51	51	1275.83	1352.51	445.80	458.21	447.87	428.87	435.60	428.72
Bier127	127	8392.76	8667.83	1066.13	9807.66	8345.9	9298.97	8290.67	8190.8

TABLE II: PERFORMANCE COMPARISON FOR RTSP

City Problem	SA	TS	GA	MA	BCO	ACO	FA	CS
25	2103.07	2067.12	448.06	420.51	440.78	415	378.09	305.6
75	1275	1314.80	818.63	757.96	1081	1087	544.36	504.3
100	4195.67	4200	894.48	840.24	843.20	1188	614.18	612
200	5843	5600.87	1296.90	1216.19	2070	2073	1559.8	1188.78