Journal of Informatics
Electrical and Electronics Engineering, 2022,
Vol. 02, Iss. 03, S. No. 004, pp. 1-9
ISSN (Online): 2582-7006

Solution of Travelling Salesman Problem based on Metaheuristic Techniques

Himanshu Kumar Mishra[1], Pawan Singh[2], Anil Tiwari[3]

[1,2,3]Amity School of Engineering and Technology, Amity University Uttar Pradesh, Lucknow, India
[1]himanshukm2301@gmail.com, [2]pawansingh51279@gmail.com, akumar3@lko.amity.edu[3]

How to cite this paper: H. K. Mishra and P. Singh, "Solution of Travelling Salesman Problem based on Metaheuristic Techniques," *Journal of Informatics Electrical and Electronics Engineering* (JIEEE), Vol. 02, Iss. 03, S No. 004, pp. 1–9, 2022.

https://doi.org/10.54060/JIEEE/002.03.004

Received: 02/10/2021
Accepted: 02/11/2021
Published: 27/11/2021

Abstract

The traveling salesman problem is a classic problem in combinatorial optimization. This problem is to find the shortest path that a salesman should take to traverse through a list of cities and return to the origin city. The list of cities and the distance between each pair are provided. It is an NP-complete problem i.e., class of computational problem for which no efficient solution algorithm has been found, presently there is no polynomial solution available. In this paper, we try to solve this very hard problem using various heuristics such as Simulated Annealing, Genetic Algorithm to find a near-optimal solution as fast as possible. We try to escape the local optimum, using these advanced heuristic techniques.

Keywords

Genetic Algorithm; Simulated Annealing; heuristic;

1. Introduction

Travelling salesman problem can be formulated as "For a given list of nodes determine the shortest tour covering all nodes and visiting each node only once in doing so." Travelling Salesman problem has many real-world applications such as drilling of Printed Circuit Boards, X-Ray crystallography, Computer Wiring, Vehicle routing, Order picking problem in warehouses. It is hard to solve since the number of possible solutions increases exponentially with the size of input. Even the naive solution which goes through all possible arrangements of nodes has a O(n!) complexity. This Problem has many variations like Symmetric Travelling Salesman Problem and Asymmetric travelling Salesman Problem. In Symmetric TSP, the distances between two nodes are same forwards and backwards, while in Asymmetric TSP the distances between two nodes may not be same in either direction. Available deterministic methods like dynamic programming, branch and bound techniques.

2. Motivation

In today's world where we have a huge amount of computational power and lot of resources. The best algorithm to solve TSP is still not invented. Scientists and Algorithmist have been working on it to improve its solution since last 50 years and still the 1-million-dollar prize money by Clay Mathematics institute of Cambridge Massachusetts is unclaimed. We being the students of Computer Science thought of solving the problem by our own methods and experiments.

3. Problem Definition

Given list of n nodes, we have to find the optimal cost of travelling n cities, where the cost between two cities is defined as the Euclidean distance, and objective is to minimize the total distance of the tour.

4. Simulated Annealing

Simulated Annealing is a simulation of original Annealing process in metallurgy used as a Heuristic Algorithm in Computer Science, to approximately calculate the global maximum/minimum of a function. When the solution set of a problem is very large, we use Simulated Annealing to jump from one set to another in order to quickly and most efficiently calculate the maximum or the minimum value in the set which is as close as possible to the best solution.

Consider the travelling salesman problem where we have to find the shortest tour covering all the cities in the country. We can have in total n! Possible tours. The search space is too large to evaluate each and every tour and select the best one. Simulated Annealing is one of the techniques to find the most suitable solution in sensible length of time for this problem and not the most optimal one. The process is adapted from annealing in metallurgy. Annealing means slow cooling. As metal cools down it becomes more capable of holding its most recently acquired structure.

We use Simulated Annealing to overcome the anomaly of getting stuck on local optimum, which by the way most of Heuristic Algorithms are prone to. In Simulated Annealing we gradually decrease the temperature from some initial point to zero. We generate a new solution using the current solution, and calculate its energy, which further determines whether or not to accept the newly generated solution. Just like the original Annealing process SA discards all the solutions with more energy than the current one at low temperature, while allowing them to proceed further at high temperature. Therefore, the algorithm's ability to perform this operation at high temperature saves it from getting stuck at local optimum.

4.1. Equations

$$\boldsymbol{P = e^{-\Delta E/kT}}$$

P: Probability function
T: Current temperature
k: Boltzmann Constant
ΔE: Difference between next and current energy states

4.2. Solving TSP using SA

- First, we randomly generate a tour of the given set of cities.

- Then we initialize the initial temperature and the cooling rate of the process.
- Next, we loop until the temperature is still greater than 1.
- We select the next tour by swapping 2 random positions from our current tour.
- Then using the probability function, we decide whether to move to the next solution or not.
- We keep track of the best solution while looping through all the tours.
- Temperature is cooled by the predetermined cooling factor in each iteration.
- Once the process ends, we have the best possible.

4.3. Simulated Annealing Pseudo Code

1. Choose an initial tour S

2. Choose a temperature $T(1) = T(0) > 0$

3. Repeat:

    a. Choose a new tour S'

    b. Let $\Delta E = E(S') - E(S)$, where $E(S)$ is energy (length) of tour S

    c. If $\Delta E \leq 0$, accept new tour i.e., $S = S'$

    d. Else if $e^{-\Delta E/kT} \geq rand(0,1)$, accept new tour

    e. Else reject new tour

f. Reduce temperature according to cooling factor

4. Until termination conditions are met

5. Genetic Algorithm

Genetic algorithm is a commonly used method which is used to solve a number of optimization problems. It is inspired from the natural process of evolution, in which each individual transfers some characteristics or attributes to the next generation, i.e. it transfers genetic information onto the next generation with some mutation. This method is used to give plausible solutions to optimization problems with large data sets which require high computational complexity to find the best answer.

5.1. Genetic description of Algorithm

Firstly, an initial population of individuals is chosen. In this population, each individual represents a possible solution to the problem. Then a new generation of the population is generated from this population by selection, crossover and mutation methods. Each new generation contains a better set of solutions than the previous generation. New generations are kept on generating until a termination condition is reached.

5.2. Use of Genetic Algorithm to solve TSP

Individual: Each individual is a possible solution for the problem, in this problem each possible solution is a tour of the cities. A tour of cities can be represented as permutation of the cities. For e.g., Suppose c1, c2, c3, c4 ... cn are cities given. Consider a permutation of these cities c3, c5, c1 ... ck. This permutation represents the tour c3 -> c5 -> c1 -> .... ck -> c3.

Population: Population is the set of individuals. In our solution we choose randomly generated permutations for the cities as our initial population.

Selection: Selection means we select two parents. For selection we use tournament selection method. We randomly select individuals from the population and choose the fittest individual from the selected individuals as parent1. Similarly, we choose the second parent.

Crossover: After selection of parents, we generate a new chromosome. For chromosome generation crossover operation is performed. In this case we perform ordered crossover.

The crossover method is described in the given image. A subset of the first parent is added to the child and the remaining values are added from the second parent.

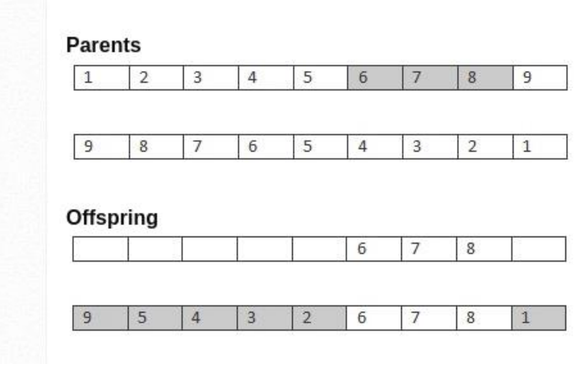

Figure 1. Ordered Crossover

5.3. GA Pseudo Code

Generate initial population of random tours
While time < 10 sec
    Select two fittest tours as parents
    Combine them to generate offspring/child tour (Ordered Crossover)
    Mutate child tour
    Calculate fitness of the child
    New generation created
End while

6. Hybrid Simulated Annealing and Genetic Algorithm

Hybrid Simulated Annealing and Genetic Algorithm is a combined Algorithm in AI to generate the best possible solution for Travelling Salesman Problem.

6.1. General description of method

Firstly, an initial population of individuals is chosen. In this population, each individual represents a possible solution to the problem. Then a new generation of the population is generated from this population by selection, crossover and mutation methods. The trick to improve the solution of TSP lies in the selection step. We call this step as the select and improve step. Each new generation contains a better set of solutions than the previous generation and the most suitable generations are improved after the selection and then sent to the next step for crossover. New generations are kept on generating until a termination condition is reached.

Basically, we are trying to improve the efficiency of solution of Travelling salesman Problem by improving the method of generating the solution, using simulated annealing as subroutine function. The Simulated annealing algorithm was also improved from previous method, by selecting the next solutions by choosing two random indices and then reversing the list of nodes present between them. This way the randomness of SA method was increased and solutions were greatly improved from previous version. For example: Let the tour initially be 1-2-3-4-5-6. After reversing nodes between 2 and 4 we have 1-4-3-2-5-6.

6.2. Use of HSAGA to solve TSP:

Individual: Each individual is a possible solution for the problem, in this problem each possible solution is a tour of the cities. A tour of cities can be represented as permutation of the cities. For e.g.: Suppose c1, c2, c3, c4 ... cn are cities given. Consider a permutation of these cities c3, c5, c1 ... ck. This permutation represents the tour c3 -> c5 -> c1 -> .... ck -> c3.

Population: Population is the set of individuals. In our solution we choose a total of 100 permutations. 98 of which are randomly generated. 1 is generated by greedy algorithm while 1 is the output of Simulated annealing process applied on given random permutations for the cities as our initial population.

Selection: Selection means we select two parents. For selection we use tournament selection method. We randomly select individuals from the population and choose the fittest individual from the selected individuals as parent1. Similarly, we choose the second parent.

Crossover: After selection of parents, we generate a new chromosome. For chromosome generation crossover operation is performed. In this case we perform ordered crossover. This process is same as described in original GA process.

The method described in reference paper; Genetic algorithm is using Simulated Annealing as subroutine. And this was done for most optimal 20 percent individuals in each and every iteration, initial temperature and cooling factor are based fitness value of the individuals (most fit and least fit individuals) simulation parameters as follows:

i. Population size: 200
ii. Probability of crossover: 0.8
iii. Probability of mutation: 0.1
iv. 4. Frozen condition for SA: 0.005

When experimented, the above-mentioned algorithm had many limitations, which we have improved upon to produce even better results. For example, keeping temperature as a function of fitness function reduced the efficiency when the coordinate points were smaller, so to improve upon this we kept the temperature constant.

Upon testing we observed that if we increased the temperature constant it produced better results but in turn it reduced the number of generations produced via GA. So, to increase the temperature we have to give up in the percentage of the population on which SA is called on. We also found that optimal results were produced when temperature was set at 10000000 (10^7) and the population percentage was set at 1-2%.

6.3. Pseudo Code

Simulated Annealing

i. Choose an initial tour S

ii. Choose a temperature T (1) = T (0) > 0

iii. Repeat:

   a. Choose a new tour S' using new procedure described

   b. Let AEE(S) - E(S), where E(S) is energy described(length) of tour S

   c. If ΔE ≤ 0, accept new tour i.e. S = S`

   d. Else if $e^{-\Delta E/ kt}$ >= rand (0, 1), accept new tour

   e. Else reject new tour f. Reduce temperature according to cooling factor

iv. Until termination conditions are met

Genetic Algorithm Pseudocode

i. **Generate** initial population of random tours

ii. Add I tour which greedily chooses the nearest city as the next city

iii. Repeat:

While time < 20 sec

Select a random tour from the current population and call simulated annealing on this tour

Select two fittest tours as parents, using tournament selection

Combine them to generate offspring/child tour (Ordered Crossover)

Mutate child tour

Propagate the fittest individual in current population onto the next generation.

Calculate fitness of the child

New generation created

End while

7. Literature Survey

According to Wikipedia, the origins of TSP is yet not known. It was first found in a handbook in 1832 without any prior mathematical statements. It was later formulated by mathematicians W.R Hamilton and Thomas Kirkman. The general form of TSP was first studied and stated by Karl Menger in the '30s as follows:

"We denote by messenger problem (since in practice this question should be solved by each postman, anyway also by many travelers) the task to find, for finitely many points whose pairwise distances are known, the shortest route connecting the points. Of course, this problem is solvable by finitely many trials. Rules which would push the number of trials below the number of permutations of the given points are not known. The rule that one first should go from the starting point to the closest point, then to the point closest to this, etc., in general does not yield the shortest route."

8. Methodology

The work was divided into phases i.e., pre–Mid Sem which has already been evaluated during the Mid Sem Evaluation and post-Mid Sem.

8.1. Pre-Mid Semester

After studying various papers on solving Travelling Salesman Problem using Simulated Annealing and Genetic Algorithm with minor modifications to improve the results. Designed a Graphical User Interface, which can take input from file in TSPLIB data format.
Example:

NAME: a280

COMMENT: drilling problem (Ludwig)

TYPE: TSP

DIMENSION: 280

EDGE_WEIGHT_TYPE: EUC_2D

NODE_COORD_SECTION

The GUI finally displays optimal tour found for the given dataset using our implemented code.

Before mid-sem we have implemented primitive versions of Simulated Annealing and Genetic Algorithm in our code.

After mid-sem we plan to improve and implement third algorithm based on new developments and research in the area of TSP using SA and GA. Also improve the GUI to take interactive input from user and display the comparison between the output of TSPLIB dataset, our code and user's code. We plan to execute users code and out code on our system and finally compare the tours obtained.

8.2. Post Mid Semester:

- Improved the basic version of simulated annealing method by using better evaluation function for generating new solutions from existing solutions, which increases the efficiency of the solutions.
- Implemented the new algorithm based on the reference material, by combining the methods mentioned in the multiple reference materials and combining the ideas from them (greedy + simulated annealing + genetic algorithm).
- Experimented with various methods, for generating new solutions, and finally settled for the best one as described in the algorithm above.
- Newly Implemented algorithm has significant improvement over the previous algorithms.
- Added extra features in GUI (Graphical User Interface), the user can select random points on the screen, and the algorithm will find optimal tour for the selected points.
- Included feature, in which user can insert his code or type code in the given window and run the algorithm on the standard data sets or user input and compare the results with best available solutions, and also solution implemented by us during the course of this project.

9. Results

Designing an interactive GUI application which takes user code as input and displays the comparison between the solution of TSP on TSPLIB dataset, on our code’s output and user's output. To study and improve the best available algorithms in terms of time and optimal solution.

10. Comparison

TSP is a dataset directory that contains some examples of data for the travelling salesperson problem. Most of these examples come from TSPLIB, a collection of traveling salesman problem datasets maintained by Gerhard Reinelt at "http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/"

Example:

NAME: eil76, **COMMENT**: 76-city problem (Christofides/Eilon), **TYPE**: TSP, **DIMENSION**: 76, **EDGE_WEIGHT_TYPE**: EUC_2D

Table 1. Comparison of ours

Data set	TSP tours			
	SA	*GA*	*HSAGA*	*Optimal*
ch130	7551.063	7456.1359	6187.0601	6110.86
eil76	614.3852	572.2340	556.035	545.388
tsp225	4838.3961	4828.9966	4052.3765	3859.0000
pr76	117795.113	113706.07	108444.0471	108159.4382
kroA100	24710.2388	23381.9883	21344.7158	21285.4431
ch150	8075.3039	8049.6781	6665.6626	6532.2809
a280	3184.9277	3148.1099	2651.12	2586.77
eil51	445.4857	439.6724	428.87	429.9833
berlin52	8328.2487	8086.3125	7544.3659	7544.3659

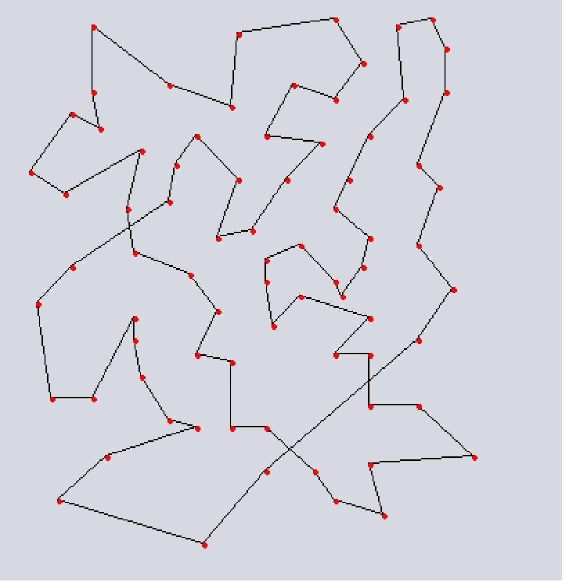

Figure2. eil76 Genetic Algorithm

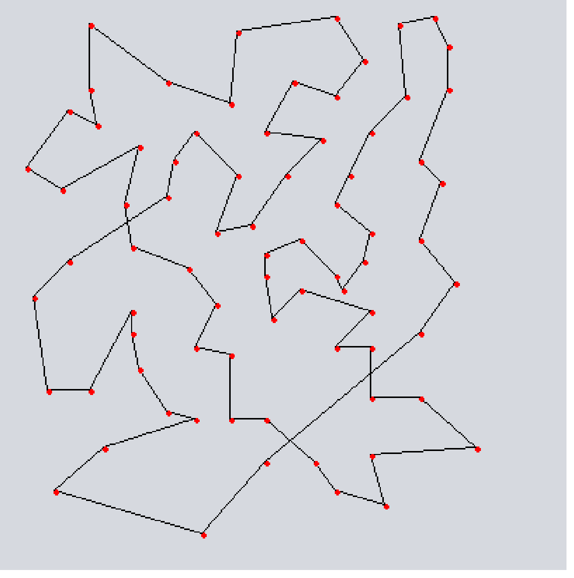

Figure 3. eil76 Simulated Annealing Algorithm

11. Conclusion

During the course of this work, we have implemented, new algorithm for solving Travelling Salesman Problem with help of references mentioned and also other sources. The algorithm which we have implemented has achieved an accuracy of more than 95% percent in most of the cases. Given the limitations of computing power and resources, we expect to achieve better results by using machines with more computational power and also take forward the work done during this semester to future projects to explore more in this domain.

References

[1]. R. R. Geetha, N. Bouvanasilan and V. Seenuvasan, "A perspective view on Travelling Salesman Problem using genetic algorithm," 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009, pp. 356-361, doi: 10.1109/NABIC.2009.5393321.

[2]. G. Zhao, W. Luo, H. Nie and C. Li, "A Genetic Algorithm Balancing Exploration and Exploitation for the Travelling Salesman Problem," 2008 Fourth International Conference on Natural Computation, 2008, pp. 505-509, doi: 10.1109/ICNC.2008.421.

[3]. J. W. Pepper, B. L. Golden and E. A. Wasil, "Solving the traveling salesman problem with annealing-based heuristics: a computational study," in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 32, no. 1, pp. 72-77, 2002. doi: 10.1109/3468.995530.

[4]. B. Golden, J. Pepper and T. Vossen, "Using genetic algorithms for setting parameter values in heuristic search" in Intelligent Engineering Systems Through Artificial Neural Networks, New York:ASME, vol. 8, pp. 239-245, 1998.

[5]. D. Johnson, C. Aragon, L. McGeoch and C. Schevon, "Optimization by simulated annealing: An experimental evaluation; Part I Graph partitioning", Oper. Res., vol. 37, no. 3, pp. 378–406, 1991.

[6]. S. P. Coy, B. L. Golden, G. C. Runger and E. A. Wasil, "See the forest before the trees: fine-tuned learning and its application to the traveling salesman problem," in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 28, no. 4, pp. 454-464, 1998. doi: 10.1109/3468.686706

[7]. J. T. Alander, "An Indexed Bibliography of Distributed Genetic Algorithms", Technical Report No. 94-1-PARA", Department of Information Technology and Production Economics, University of Vaasa, Finland, 1997.

[8]. T. Starkweather, D. Whitley, and K. Mathias, "Optimization using distributed genetic algorithms," in Parallel Problem Solving from Nature, Berlin/Heidelberg: Springer-Verlag, 2006, pp. 176–185.

[9]. A. Simões, E. Costa, "Using genetic algorithms to deal with dynamic environments: a comparative study of several approaches based on promoting diversity", The Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, CA: Morgan Kaufmann, New York, 2002.

[10]. M. Misra and P. Singh, "Energy Optimization for Smart Housing Systems," Journal of Informatics Electrical and Electronics Engineering, vol. 01, no. 5, pp. 1–6, 2020.

[11]. S. Jain, "Operational cost minimization of grid connected microgrid system using firefly technique," Journal of Informatics Electrical and Electronics Engineering (JIEEE), vol. 1, no. 2, pp. 1–26, 2020.

[12]. N. Srivastava, U. Kumar, and P. Singh, "Software and Performance Testing Tools," Journal of Informatics Electrical and Electronics Engineering (JIEEE), vol. 2, no. 1, pp. 1–12, 2021.

[13]. A. Singh and P. Singh and A. K. Tiwari, "A Comprehensive Survey on Machine Learning," Journal of Management and Service Science, vol. 1, no. 1, pp. 1–17, 2021.