International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 8 Issue 2 February - 2024, Pages: 1-6

Traveling Salesman Problem: A Comparative Analysis of Three Solving Strategies

Asaad Naser Hussein Mzedawee

Department of Statistics, College of administration and economics, University of Al-Qadisiyah, Iraq
Correspondent Author: asaad.nasir@qu.edu.iq

Abstract*— In this paper, we present three methods to solve the traveling salesman problem, which is the Np-hard problem. In fact, this problem was previously solved using classical methods, which suffered from shortcomings and unsatisfactory solutions. This problem was sought to be solved using modern optimization algorithms by three optimization algorithms were used to solve this problem: (Genetic Algorithm, Brute Force, Dynamic Programming method). Performance was compared in terms of time and speed, and the dynamic programming method was the best in terms of performance. The numerical results in the optimization process were obtained by Python programming.*

Keywords— Optimization technique, Traveling Salesman Problem, Genetic Algorithm, Brute Force method, Dynamic Programming method .

1. INTRODUCTION

The traveling salesman problem TSP is a classic tour problem in which a salesman must determine the best route through his region, stopping only once and returning to the starting point. Due to the combinatorial complexity of the TSP, estimated or heuristic measurements are applied almost often. The TSP's schematic and mathematical structure is a graph with attributes like nodes, edges, vertices, etc. Cities in the problem are nodes and connected by vectors called edges, which have costs like distance, time, or other attributes. For a weighted network G with n vertices representing cities, the TSP challenge is to determine the cycle of minimal costs that visits each vertex exactly once, [1-4]. The travelling salesman problem aims to discover the shortest path between all cities and the starting point given a set of cities and their distances. Though simple to declare, the assertion is harder to solve. Due of its large search area, the Travelling Salesman Problem is NP-hard and cannot be solved in polynomial time. This is one of the most fundamental computer science challenges today. The travelling salesman problem is used in many fields. Its uses include vehicle routing, microchip manufacturing, GSM packet routing, PCB drilling, etc [5-7]. If we have n cities, we can find (n - 1)! alternative routes to cover them all. Travelling salesmen must find the shortest path. There are a great many approaches that have been created for the purpose of solving TSPs. There is a highly promising direction that can be indicated by evolutionary algorithms (EAs) [8]. The concepts that underpin them are derived from the fields of genetics and natural selection. There are many different ways that an evolutionary algorithm might be utilised to address problems. It is notably useful for solving tough optimisation problems, which are challenges for which typical optimisation approaches are less effective. On the other hand, general problem-independent EAs are typically inefficient when it comes to tackling TSPs, particularly large TSP challenges. There have been many different ways proposed in order to enhance additional EAs for TSPs. In addition to these, the development of TSP-specific operators, the incorporation of local searches, and the preservation of population variety are all considered to be potential approaches to the problem of TSPs. It has been demonstrated that methods that incorporate domain-specific local search methods into EAs, also known as memetic algorithms, are also effective heuristics for solving TSPs [9–14]. Preventing the early convergence of TSP can be achieved through the implementation of specific mechanisms that have been developed and designed for this purpose. Several factors that, when combined with EAs, produce higher-quality solutions to TSPs have been the subject of research. As an illustration, Reinelt (27) provided empirical evidence that the edge exchange search algorithm exhibited a markedly higher efficiency level than both the node reinsertion and node exchange algorithms. In designing recombination operators, the preservation of edges takes precedence over the placement of nodes, according to Whitely et al. [15 Furthermore, it is recommended to focus more on the techniques used for edge fusion and preservation . The current study deals with three optimization algorithms (the genetic algorithm and the dynamic algorithm, in addition to the Brute Force algorithm, which will be compared in their approach to arrive at the optimal solution to the traveling salesman problem, see figure 1.

International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 8 Issue 2 February - 2024, Pages: 1-6

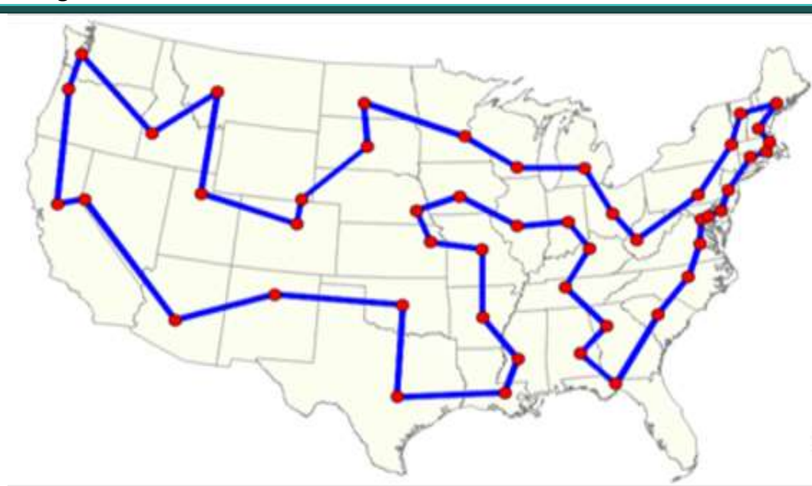

Figure 1: traveling salesman problem TSP

2. PROBLEM STATEMENT

The challenge to identify the shortest possible route that can visit every city exactly once and then return to the beginning point. Given a collection of cities and the distance between each pair of cities, the problem is to determine the shortest possible route. As we can take note of the distinction between the TSP and the Hamiltonian Cycle. Discovering whether or not there is a tour that stops in each and every city exactly once is the objective of the Hamiltonian cycle issue. In this case, we are aware that the Hamiltonian Tour exists (due to the fact that the graph is complete), and in fact, there are a great number of tours of this kind; the challenge is to determine the minimal weight. It is possible to formulate this in a mathematical form as follows:

$$min_{path} \sum_{i=1}^{n-1} d_{i,i+1} + d_{n,1}$$

Subject to:

Path ={1,2,…,n} is a permutation of the set of cities {1,2,…,n}
$d_{i,j}$ is the distance between cities i &j
$d_{i,j} = d_{j,i}$ for all cities i &j

3. APPLICATION OF TRAVELING SALESMAN PROBLEM

Travelling salesman problem is a widely common problem to answer because TSP may be applied to a wide variety of situations that occur in the real world. As an illustration, certain instances of the problem of vehicle routing can be modeled as a problem involving a travelling salesman. In this situation, the challenge is to determine which clients should be served by which vehicles and the minimal number of vehicles that are required to do so for each individual consumer. There are numerous variants of this problem, one of which is determining the shortest amount of time necessary to attend to all of the clients. As the TSP, we are able to find solutions to some of these issues. There is also the possibility of modeling the problem of computer wiring as a TSP. Each of the modules that we have is equipped with a certain amount of pins. It is necessary for us to link a portion of these. Arrange pins with wires so that each pin has a maximum of two attached wires and that the length of the wire is kept to a minimum. Plate, Lowe, and Chandrasekaran discovered an application for their work, which is referred to in [16]; it involves the overhauling of gas turbine engines in aircraft. Nozzle-guide vane assemblies, which are comprised of nozzle guide vanes that are attached to the circumferential, are situated at each step of the turbine's operation in order to guarantee a consistent flow of gas. A symmetric TSP can be used to model the optimization of the positioning of the vanes in order to reduce the amount of fuel that is consumed. TSP also refers to the process of scheduling jobs on a single machine, taking into account the amount of time required for each job as well as the amount of time required to prepare the machine for each job [17-19]. We work hard to reduce the total amount of time required to process each task. It is necessary for a robot to carry out a wide variety of tasks in order to finish a procedure. We have precedence limitations in this application, as opposed to the scheduling of jobs on a computer, which happens in other applications.

4. METHODOLOGY

The methodology used in this work consists of four basic steps, as follows:

1- **Problem generation:** We generate hypothetical traveling salesman problems of different sizes and distributions to cover all possible scenarios of the problem, meaning that the generation of the distance matrix *D*[i, j] will be random, that is the distance matrix D .

2- **Implementation of optimization algorithms:** We will apply the proposed optimization algorithms (Genetic Algorithm, Brute Force, Dynamic Programming method) to the traveling salesman problem using Python, and the requirements of each algorithm will be configured according to its workflow.

3- **Performance evaluation:** The performance of optimization algorithms is evaluated according to their computational complexity, where the time taken to run each algorithm is compared, in addition to comparing the solutions obtained by each algorithm.

5. NUMERICAL RESULTES AND DISCUSSION

After running the optimization process represented by implementing the three algorithms, the table below included the information obtained:

Table 1: Numerical results by running the optimization approach

Algorithm	Minimum Distance	Runtime (s)	Robustness to Noise	Robustness to Incompleteness
Brute Force	85	3.67	No	No
Dynamic Programming	79	0.008	Yes	Yes
Genetic Algorithm	227	0.092	Yes	No

As shown in Table 1, the optimal solution was obtained using the (Brute Force algorithm) with the shortest distance, but it had the highest level of computational complexity, in addition to taking a running time of 3.67 seconds. As for the dynamic programming algorithm, it achieved near-perfect solutions with an average running time of 0.008 seconds with less computational complexity, as it demonstrated the strength of its approach in dealing with noisy distance information that takes a pattern from annoying to incomplete. Despite this, it works perfectly in determining the optimal solution under all circumstances. As for the genetic algorithm, as is clear from the numerical results, its chances of dealing with incomplete distances were unsatisfactory, even though it is computationally efficient. In general, the results indicate that the most appropriate approach to solving the traveling salesman problem was the dynamic programming algorithm, as it had the greatest share of computational efficiency and robustness and its challenge to conditions of noise or incomplete data.

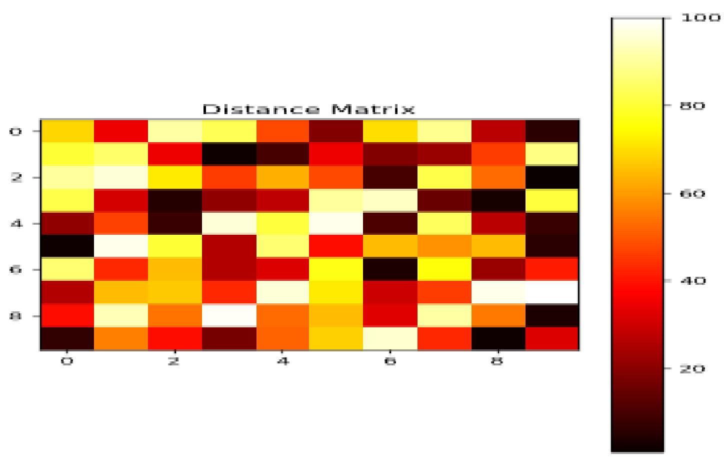

Figure 2: the distance matrix for a TSP instance with 10 cities

As shown in Figure 2, the distance matrix for the traveling salesman problem is represented within 10 cities, where dark colors indicate distances that are large. Also, figure 3 shows the optimal path found by the dynamic programming algorithm , with a minimum distance of 79. Figure 4 clearly shows the optimal path found by the brute force algorithm, with a distance of no less than 85 compared to the genetic algorithm, and as shown in Figure 5, it obtained the optimal path with a distance of no less than 227.

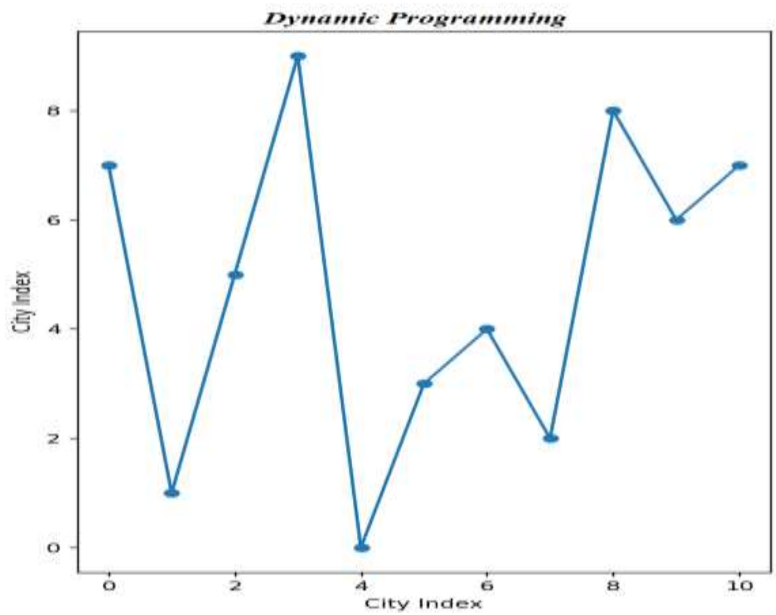

Figure 3: the distance matrix for a TSP instance with 10 cities

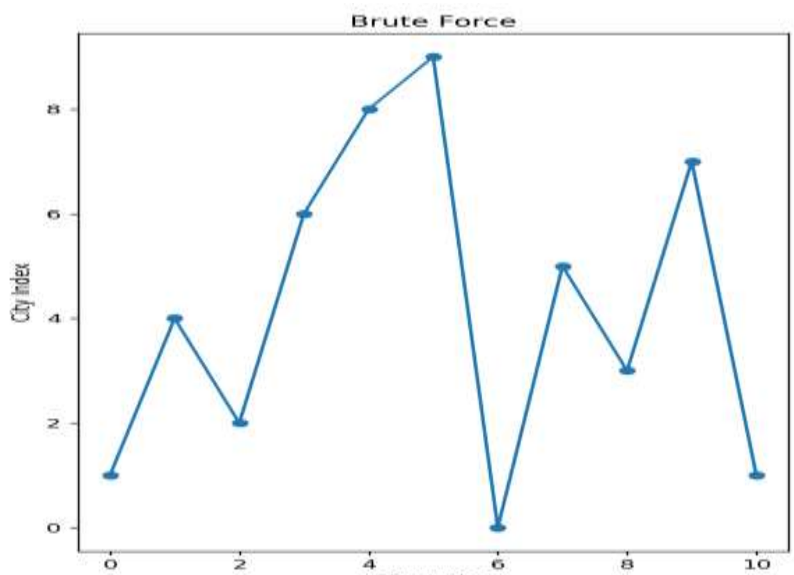

Figure 4: the distance matrix for a TSP instance with 10 cities

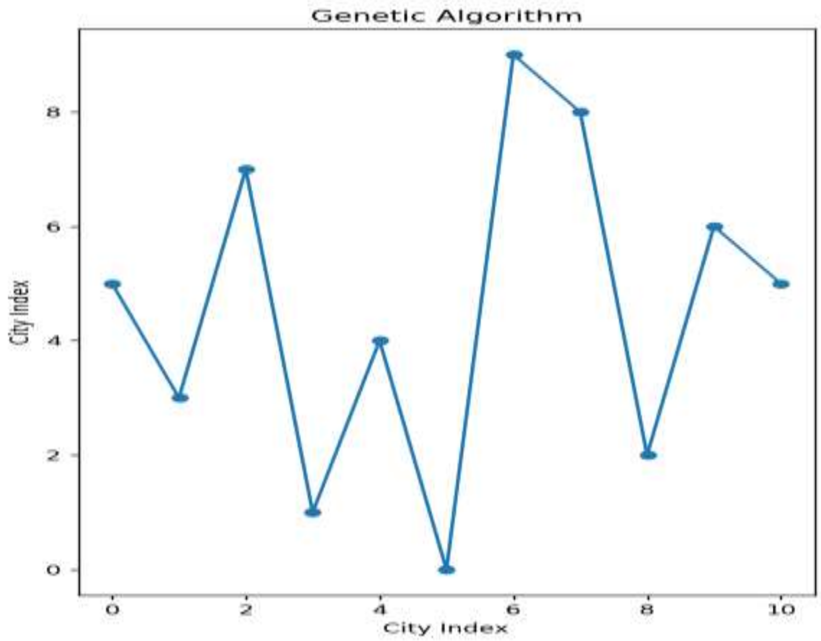

Figure 5: the distance matrix for a TSP instance with 10 cities

6. CONCLUSION

The solution to the traveling salesman problem is investigated by three algorithms. In fact, the dynamic programming algorithm outperformed the brute force algorithms and the genetic algorithm, as it achieved near-perfect solutions with less computational complexity compared to the performance of the other two algorithms.Moreover, the dynamic programming algorithm had a more robust approach to dealing with incomplete or noisy information. Finally, the results obtained highlight the scientific implications for practitioners and researchers in solving this type of problem and open other horizons for research paths that would support improvement and efficiency techniques in solving the traveling salesman problem or similar ones.

7. ACKNOWLEDGMENT

I am pleased to extend my thanks and gratitude to the management and editorial board of the journal for their support in publishing this paper

8. REFERENCES

[1] Isoart, N., & Régin, J. C. (2019). Integration of structural constraints into tsp models. In Principles and Practice of Constraint Programming: 25th International Conference, CP 2019, Stamford, CT, USA, September 30–October 4, 2019, Proceedings 25 (pp. 284-299). Springer International Publishing.

[2] Mamano, N., Efrat, A., Eppstein, D., Frishberg, D., Goodrich, M., Kobourov, S., ... & Polishchuk, V. (2019). Euclidean tsp, motorcycle graphs, and other new applications of nearest-neighbor chains. arXiv preprint arXiv:1902.06875.

[3] Sultana, N., Chan, J., Sarwar, T., & Qin, A. K. (2022). Learning to optimise general TSP instances. International Journal of Machine Learning and Cybernetics, 13(8), 2213-2228.

[4] Crişan, G. C., Nechita, E., & Simian, D. (2021). On Randomness and Structure in Euclidean TSP Instances: A Study With Heuristic Methods. IEEE Access, 9, 5312-5331.

[5] Zhang, Z., Zhang, Z., Wang, X., & Zhu, W. (2022, June). Learning to solve travelling salesman problem with hardness-adaptive curriculum. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 8, pp. 9136-9144).

[6] Dahiya, C., & Sangwan, S. (2018). Literature review on travelling salesman problem. International Journal of Research, 5(16), 1152-1155.

[7] Syambas, N. R., Salsabila, S., & Suranegara, G. M. (2017, October). Fast heuristic algorithm for travelling salesman problem. In 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA) (pp. 1-5). IEEE.

[8] Bremermann, H. J., Rogson, M., & Salaff, S. (1965). Search by evolution. Biophysics and Cybernetic Systems, 157-167.

[9] Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report, 826(1989), 37.

[10] Zhang, P., Wang, J., Tian, Z., Sun, S., Li, J., & Yang, J. (2022). A genetic algorithm with jumping gene and heuristic operators for traveling salesman problem. Applied Soft Computing, 127, 109339.

[11] Gutiérrez-Aguirre, P., & Contreras-Bolton, C. (2024). A multioperator genetic algorithm for the traveling salesman problem with job-times. Expert Systems with Applications, 240, 122472.

[12] Purusotham, S., Jayanth, T., Vimala, T., & Ghanshyam, K. (2022). An efficient hybrid genetic algorithm for solving truncated travelling salesman problem. Decision Science Letters, 11(4), 473-484.

[13] Alridha, A. H., & Al-Jilawi, A. S. (2022). Solving NP-hard problem using a new relaxation of approximate methods. International Journal of Health Sciences, 6, 523-536.

[14] Zheng, J., Zhong, J., Chen, M., & He, K. (2023). A reinforced hybrid genetic algorithm for the traveling salesman problem. Computers & Operations Research, 157, 106249.

[15] Reinelt, G. (2003). The traveling salesman: computational solutions for TSP applications (Vol. 840). Springer.

[16] Gerard Reinelt. The Traveling Salesman: Computational Solutions for TSP Applications. Springer-Verlag, 1994.

[17] Mosayebi, M., Sodhi, M., & Wettergren, T. A. (2021). The traveling salesman problem with job-times (tspj). Computers & Operations Research, 129, 105226.

[18] Alridha, A. H. (2023). OPTIMIZATION ALGORITHMS FOR PROJECTILE MOTION: MAXIMIZING RANGE AND DETERMINING OPTIMAL LAUNCH ANGLE. Journal of Fundamental Mathematics and Applications (JFMA), 6(2), 176-187.

[19] Alridha, A. H. (2023). Efficiency and Accuracy in Quadratic Curve Fitting: A Comparative Analysis of Optimization Techniques. Indonesian Journal of Applied Mathematics, 3(2), 8-14.