Proceedings of the Second International Conference on Edge Computing and Applications (ICECAA 2023)
IEEE Xplore Part Number: CFP23BV8-ART; ISBN: 979-8-3503-4757-9

Traveling Salesman Problem with Ant Colony Optimization

V. Murugananthan
School of Computing, Asia Pacific University of Technology & Innovation (APU), Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia.
mail_muru@yahoo.com

Mahmoud Yehia Emam Selim Rehan
School of Computing, Asia Pacific University of Technology & Innovation (APU), Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia.
mahmoudyehia420@gmail.com

R.Srinivasan
Professor
Department of Computer Science and Engineering, Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Chennai
srinivasanrajkumar@gmail.com

M.Kavitha
Professor
Department of Computer Science and Engineering, Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Chennai
kavitha@veltech.edu.in

R.Kavitha
Professor,
Department of Computer Science and Engineering, Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Chennai
rkavitha@veltech.edu.in

2023 2nd International Conference on Edge Computing and Applications (ICECAA) | 979-8-3503-4757-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICECAA58104.2023.10212262

***Abstract* – An application of Ant Colony Optimization (ACO) to the Travelling Salesman Problem (TSP) is presented in this research study. Ant Colony Optimization (ACO) is a novel technique for combinatorial optimization practitioners. Because ACO is based on the behavior of ant colonies, it has a significant advantage and a widely dispersed calculation mechanism. Finding optimization problems is rather easy using ACO. The "problem of the travelling salesman" is one of the most popular illustrations of combinatorial optimization. This study has used the Ant Colony Optimization (ACO) Algorithm to the Travelling Salesman Problem (TSP) in this work. This study also discusses about the most important modifications that might be made to the Ant Colony algorithm to improve its suitability for resolving the Travelling Salesman Problem (TSP) and ensuring the best results.**

Keywords—Traveling Salesman Problem (TSP), Ant Colony Optimization (ACO), Pheromones, Rho, Optimal Path

I. INTRODUCTION

Finding the shortest distance is the best answer to the Traveling Salesman Problem (TSP) of stopping in every city at least once. This issue has to be fixed as soon as possible. When there are only roughly 5 cities to visit, the TSP is easily solved using the deterministic technique.

However, once the number of towns for the salesman to visit increased to 10, 20, or even a hundred, it became difficult to discover an optimal strategy to solve the TSP in an acceptable amount of time. As a result, TSP is classified as an NP-hard problem since it is hard to solve by looking for the best solution.

If the number of cities the salesman needs to visit increases to 10, 20, or even 100, it becomes difficult to find an ideal solution to the TSP in a reasonable amount of time. As a result, TSP is classified as an NP-hard issue due to the difficulty associated with finding a solution via the route of least resistance.

The outline for this paper looks like this: Section 2 includes a summary and analysis of previous studies that have dealt with this subject. This research's materials and procedures are detailed in Section 3. You'll find the details of how the algorithm works, including its goal, algorithm, and parameters, in section 4. The outcomes of our parameter comparisons and the algorithmic improvements we made are presented in section 5 of this research. The conclusion is presented in Section 4.

II. LITERATURE REVIEW

In recent years, the use of ACO to resolve the TSP has generated a number of scholarly articles. Some studies combined ACO with modifications like the Genetic Algorithm (GA), the Held-Karp low bound, and even dimensional ACO. The contribution of ACO to the solution of TSP will be demonstrated in this study.

The issue is addressed by Petra T. and Vladimr H [1], who employ ACO alongside a tweaked local search. Using a configurable number of cities, the authors compared this approach to the Cplex optimizer to determine which one provided the best result in the allotted time at the lowest possible cost. The outcomes of both experiments demonstrate that the proposed system outperforms the Cplex optimizer. Consequently, the ACO algorithm is optimal for resolving the TSP.

A further suggestion made by Pragya and Pratyush [2] was to employ Dimensional Ant Colony Optimization (DACO) to solve the Travelling Salesman Problem (TSP). The author compared the proposed method against regular ACO, MST-ACO, PSO without C3, and C3D PSO. The results showed that the DACO provided the best possible answer, as it required the least amount of time and had the smallest relative error compared to the others.

In addition, Lijie et al. [3] apply an enhanced ACO with the Held-Karp low limit (HK-ACO) to solve TSP optimally. The author tested whether HK-ACO produces the best results by using the same number of cities but varying the number of iterations, the dimensions, and the length of tours. The author concludes that HK-ACO is the superior method for TSP after comparing it to ASrank.

Authorized licensed use limited to: UNIVERSITAS GADJAH MADA. Downloaded on March 23,2024 at 07:22:09 UTC from IEEE Xplore. Restrictions apply.

In addition, a combination of ACO and MST was proposed for TSP by Yuren and Wei [4]. The effectiveness of the suggested system is demonstrated in terms of how quickly the solution is generated and how accurately the findings are displayed.

In order to address TSP, Chang et al. [5] also contrasted the estimate of distribution ACO (ED- ACO) approach with the standard ACO method. The outcomes demonstrated that ED-ACO is superior to ACO in terms of efficiency and effectiveness.

As a result of ACO, superior algorithms can be developed. To address the TSP, most recent publications have turned to upgraded ACO. In order to make ACO generate better optimal solutions, additional study is needed.

III. MATERIALS AND METHODS

Seeing the natural behavior of ants was the motivation for the ant colony optimization algorithm, therefore knowing this is essential to fully grasping the concept.

Ants follow a very standard protocol when looking for food. The ants' ability to coordinate and share information in order to locate the best route to food sources relies on these simple guidelines. Pheromone trails are a major factor in ant behavior. This is the sole way ants can signal to one another that they have discovered a new food supply and the general direction it is located in. Other ants can then follow these pheromone trails to the source of the meal. However, ants are strong followers of pheromone trails. Whether the ant decides to continue along the current path or take a different one depends on the relative strength of the signals they've received along the way. Thus, the route with the most potent pheromones is the one that is typically chosen.

Pheromone trails gradually fade away over time until the path is used by other ants, which strengthens the trail. Consequently, these abandoned ant paths will eventually lead to a dead end since the ants no longer use them to navigate to a food source. Because of this, ants are free to explore new territories in search of food. It's also interesting to note that different ant species frequently have distinct methods of delivering pheromones.

IV. ALGORITHM IMPLEMENTATION

A. Purpose

Marco Dorigo first suggested the ant colony optimization (ACO) method in 1992. This method optimizes ant colonies by mimicking the foraging behavior of individual ants. Swarm intelligence, of which ACO is a subset, is an approach to problem-solving that relies on decentralization, behavior collection, and AI. ACO's real-world solutions include solutions to combinatorial optimization problems. Take the well-known "travelling salesman problem," for instance. Other issues with similar criteria are also addressed using this method.

The ability of this ant colony optimization algorithm to adapt to new circumstances is a significant benefit when compared to other optimization algorithms. Being flexible enough to adjust to new circumstances on the fly is a major factor in its success.

B. Algorithm

The entire process of ant colony optimization may be broken down into just a few simple phases. In the first phase, pheromone trails are laid and the ants in the colony make their decisions individually based on the signals they receive. When the ants find a suitable pheromone trail, they will follow it and, depending on its quality, either use it or reinforce it. The boundaries and routes between the cities in the traveling salesman problem are the primary focus. The pheromone will dissipate from each portion of the trail at a different rate once all the ants have chosen a course and left their pheromone deposits. The process is then repeated as often as is necessary to locate the best solution.

1. Initialize parameters:
   - colony_size: the number of ants in the colony
   - max_iterations: the maximum number of iterations
   - rho: pheromone evaporation rate
   - alpha: pheromone importance factor
   - beta: heuristic information importance factor
2. Initialize the pheromone matrix:
   - Set all pheromone values to a small constant value
3. Repeat until the maximum number of iterations is reached:
   3.1. Create a new empty ant tour for each ant in the colony
   3.2. Repeat for each ant in the colony:
      3.2.1. Choose the next city to visit based on the probability calculated using the pheromone matrix and heuristic information
      3.2.2. Add the selected city to the ant's tour
      3.2.3. Update the set of unvisited cities
   3.3. Update the pheromone trails:
      3.3.1. Evaporate pheromone on all edges in the pheromone matrix using the evaporation rate (rho)
      3.3.2. Deposit pheromone on the edges of each ant's tour:
         - Calculate the amount of pheromone to deposit based on the tour length
         - Update the pheromone matrix with the deposited amount of pheromone
   3.4. Find the best ant tour with the shortest distance
      - Identify the ant with the shortest tour distance
   3.5. Update the best tour if a shorter tour is found
4. Return the best tour found

Figure 1: Steps of algorithm implementation

i. Constructing a path

When foraging, a specific ant will typically stick to a path marked by a particularly potent pheromone trail. However, a small element of randomness is essential for the ants' decision-making process so that it can evaluate various courses besides the ones it has recently followed. The ants' decision-making process is helped along by a heuristic value. The distance along each edge between each pair of coordinates is the heuristic to use in the TSP. The likelihood that an ant will pick an edge increases with its length.

$$p_{ij}^{k} = \frac{[t_{ij}]^{\alpha} \cdot [n_{ij}]^{\beta}}{\sum_{l \in N_i^k} [t_{il}]^{\alpha} \cdot [n_{il}]^{\beta}}$$

Here's an equation that works out the odds of just one factor entering the mix. The pheromone concentration between points I and j is denoted by tij in the equation, and its approximate value is nij. Parameters that influence the proportion of pheromone trail to heuristic data utilized in making the choice of component. If TSP were a graph, each coordinate would stand in for one city. Therefore, an ant will choose its next settlement depending on its distance from the current settlement and the amount of pheromones along the route between settlements. Hacker efficiency enhanced by simulating an ant colony (2021)

Authorized licensed use limited to: UNIVERSITAS GADJAH MADA. Downloaded on March 23,2024 at 07:22:09 UTC from IEEE Xplore. Restrictions apply.

ii. Local Pheromone Update

When an ant completes a path successfully, the local pheromone updating mechanism is activated. This is done to simulate the pheromone trails that actual ants leave behind when they discover a new food source.

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/C^{k} & \text{if component}(i,j) \text{ was used by ant} \\ 0 & \text{Otherwise} \end{cases}$$

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

According to this formula, the entire distance between two points is equal to Ck, which can be thought of as the total cost of the solution. Typically, Q is set to 1, but it can be changed to alter the pheromone emission. The total amount of pheromone to be deposited on the component can be calculated by adding Q/Ck for each solution that employs the component (i, j) (i, j). [6] Hacker efficiency enhanced by simulating an ant colony (2021)

iii. Global Pheromone Update

Once pheromones have dissipated into their constituent parts, we have reached the global pheromone update. After each iteration of the algorithm, the global pheromone update is implemented if all of the ants have successfully used a path through the local update rule.

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij}$$

When I and j are two states, the amount of pheromone presents on the pathways connecting them is represented by the following equation. However, is the variable that changes how much water evaporates. Hacker efficiency enhanced by simulating an ant colony (2021)

iv. Optimization

Many variants of ant colony optimization, such as elitist and MaxMin, are possible within this algorithm. These modifications are expected to outperform the regular ACS in some settings and difficulties.

Elitist

When the local pheromone is updated in elitist ant colony optimization systems, the current best ant or the best ant globally deposits more pheromone than is necessary. This would encourage the entire colony to start over and find a solution that has a proven track record of success. As a result, search results will improve.

MaxMin

MaxMin's approach resembles the elite ant colony optimization algorithm in that it favors a higher quality ranking trail. With MaxMin, only the best current or best overall solution—and not the stronger path—is permitted to leave a pheromone trail. MaxMin adds the requirement that the pheromone trails be kept within a small area, between two preset sites.

This is done so that there is only a small window of variation in the pheromone levels on trails, preventing the system from settling too quickly on a sub-optimal option. Hacker efficiency enhanced by simulating an ant colony (2021)

This quality is replicated in ant colony optimization by adjusting the quantity of pheromone emitted along a route according to how effectively the scored paths have been traversed. In the classic "traveling salesman problem," pheromones are left along the routes that one hypothetical ant took between cities.

C. Parameters

To execute an ACO, it is necessary to meet a number of criteria. When dealing with various optimization issues, these parameters change. Different stages of optimization may have different parameters due to varying levels of problem fear and requirements. These settings are determined through trial and error. Similarly, a traveling salesman problem can be solved by adjusting a few key variables. Factors include the size of the colony, the number of steps the ants take, the rate of pheromone evaporation, the amount of pheromone deposited, and the concentration of the pheromones to begin with.

```
def __init__(self, mode='ACS', colony_size=10, elitist_weight=1.0, min_scaling_factor=0.001, alpha=1.0, beta=3.0,
             rho=0.1, pheromone_deposit_weight=1.0, initial_pheromone=1.0, steps=100, nodes=None, labels=None):
    self.mode = mode
    self.colony_size = colony_size
    self.elitist_weight = elitist_weight
    self.min_scaling_factor = min_scaling_factor
    self.rho = rho
    self.pheromone_deposit_weight = pheromone_deposit_weight
    self.steps = steps
    self.num_nodes = len(nodes)
    self.nodes = nodes
```

Figure 2: Code of ACO

```
_steps = 50
_nodes = [(random.uniform(-400, 400), random.uniform(-400, 400)) for _ in range(0, 15)]
```

Figure 3: Code of ACO

Parameters for ant colony optimization are shown in Figures 2 and 3 below. The 'colony size' in this optimization procedure stands for the total number of ants. The term "initial pheromone" refers to the initial quantity of pheromones released by the ants, while the terms "rho" and "pheromone deposited weight" refer to the rate of pheromone evaporation after it has been deposited on the path by the ants, respectively. In conclusion, the value of "steps" represents the entire distance covered by the ants.

V. Results & Comparison

We have evaluated and contrasted the ACS, Elitist, and MaxMin ACO methods. Each city in this experiment is assigned a random set of coordinates, and there will be a total of 15.

Therefore, we will evaluate the algorithm's overall distance against the variations in colony size and rho. If we run the code five times with varying colony sizes and rho values, we can calculate the mean total distance.

The first table is a comparison of the total distance between each ACO method with the colony size set to 5, 10, or 15 and the rho held constant at 0.1. Based on the data in the table, we know that a larger colony size will provide a better solution.

Authorized licensed use limited to: UNIVERSITAS GADJAH MADA. Downloaded on March 23,2024 at 07:22:09 UTC from IEEE Xplore. Restrictions apply.

Table 1. Comparison of ACS, Elitist and Max min with colony size

Optimization Techniques	Colony Size	Average of total distance
ACS	5	3042.792
	10	2767.742
	15	2691.312
Elitist	5	3046.888
	10	2764.368
	15	2662.434
MaxMin	5	3015.144
	10	2700.584
	15	2632.076

Also, rather than starting with a colony size of 5, which would be the default for most analyses, we can use a difference of 0.1, 0.5, or 0.9 to establish the relationship between rho and average distance. Using the data in the table below, we can see that the average total distance traveled decreases as rho increases.

Table 2. Comparison of ACS, Elitist and max min values with Rho

Optimization Techniques	Rho	Average of total distance
ACS	0.1	3042.792
	0.5	2966.752
	0.9	2908.996
Elitist	0.1	3046.888
	0.5	2949.538
	0.9	2875.524
MaxMin	0.1	3015.144
	0.5	2972.318
	0.9	2836.936

Finally, by maintaining constants like colony size and rho, we may draw reliable comparisons between the three methods. Based on the data, we know that the MaxMin method will produce the best results with a colony size of 15 and a correlation coefficient of 0.9. Simply put, a smaller average total distance corresponds to a larger colony size and higher rho, hence a larger colony will produce better results.

Table 3. Details of the optimization with colony size and Rho

Optimization Techniques	Colony Size	Rho	Average of total distance
ACS	15	0.9	2879.382
Elitist	15	0.9	2876.698
MaxMin	15	0.9	2779.848

Due to allowing the best current or best global solution to leave a pheromone trail, these findings show that MaxMin is superior than the other two variants of the algorithm. In addition, it demonstrates that early convergence around sub-optimal solutions can be prevented by limiting the range between the pheromones' maximum.

VI. CONCLUSION

From these findings, we infer that the MaxMin optimization strategy yields the best possible outcome. Regardless of the variation in colony size or rho, the average of total distance created by MaxMin is still the smallest when compared to Elitist and ACS.

This study has used the Ant Colony Optimization (ACO) algorithm to implement a solution to the Traveling Salesman Problem over 15 different locations. Our code was successfully implemented, demonstrating a high rate of accuracy, and achieving the goals we set out to attain at the outset of the project.

Based on the findings, it is crucial to further refine the Ant Colony Optimization (ACO) algorithm in order to effectively address the specified challenge. The development of this algorithm offers a potential means of increasing the accuracy of outcomes for the challenge at hand and of solving problems of greater complexity. The development of the algorithm is crucial for finding effective solutions to difficult problems and demonstrating a high degree of precision in one's work.

REFERENCES

[1] Petra Tomanová and Vladimír Holý. (2020). Ant Colony Optimization for Time-Dependent Travelling Salesman Problem. In Proceedings of the 2020 4th International Conference on Intelligent Systems, Metaheuristics Swarm Intelligence (ISMSI '20). Association for Computing Machinery, New York, NY, USA, pp. 47–51.

[2] Pragya, M. Dutta and Pratyush. (2015) "TSP Solution Using Dimensional Ant Colony Optimization," in 2015 Fifth International Conference on Advanced Computing & Communication Technologies (ACCT), Haryana, India, pp. 506-512.

[3] S. Ju, Y. Zhang and L. Li, (2008) Improved Ant Colony Optimization for the Traveling Salesman Problem. Intelligent Computation Technology and Automation, International Conference. null. pp. 76-80.

[4] Z. Yuren and L. Wei, (2010) "An Effective Hybrid Ant Colony Algorithm for Solving the Traveling Salesman Problem," in *Intelligent Computation Technology and Automation, International Conference on*, Changsha, Hunan, China, pp. 497-500.

[5] X. Jun, X. Chang and C. Huiyou, (2009) "Ant Colony Optimization Based on Estimation of Distribution for the Traveling Salesman Problem," in *2013 International Conference on Computing, Networking and Communications (ICNC)*, Tianjian, China, pp. 19-23.

[6] Theprojectspot.com. 2021. Ant Colony Optimization For Hackers. [online] Available at: https://www.theprojectspot.com/tutorial-post/ant-colony-optimization-for-hackers/10

[7] G. Gutin, A.P. Punnen, The Traveling Salesman Problem and its Variations, Springer, Boston, 2007.

[8] Y. Wang, J.B. Remmel, A binomial distribution model for the traveling salesman problem based on frequency quadrilaterals, J. Graph. Algorithms Appl. 20 (2) (2016) 411–434.

[9] Muren, J.J. Wu, L. Zhou, Z.P. Du, Y. Lv, Mixed steepest descent algorithm for the traveling salesman problem and application in air logistics, Transp. Res. E 126 (2019) 87–102.

[10] H. Savuran, M. Karakaya, Efficient route planning for an unmanned air vehicle deployed on a moving carrier, Soft. Comput. 20 (7) (2016) 2905–2920.

[11] P. Baniasadi, M. Foumani, K. Smith-Miles, V. Ejov, A transformation technique for the clustered generalized traveling salesman problem with applications to logistics, European J. Oper. Res. 285 (2) (2020) 444–457.

[12] [R.M. Karp, On the computational complexity of combinatorial problems, Networks 5 (3) (1975) 331–333.

Authorized licensed use limited to: UNIVERSITAS GADJAH MADA. Downloaded on March 23,2024 at 07:22:09 UTC from IEEE Xplore. Restrictions apply.

[13] S. Climer, W.X. Zhang, Cut-and-solve: An iterative search strategy for combinatorial optimization problems, Artificial Intelligence 170 (8–9) (2006) 714–738.
[14] M. Held, R.M. Karp, A dynamic programming approach to sequencing problems, J. Soc. Ind. Appl. Math. 10 (1) (1962) 196–210.
[15] J.J. Grefenstette, R. Gopal, B. Rosmaita, D. Van Gucht, Genetic algorithms for the traveling salesman problem, in: Proceedings of the First International Conference on Genetic Algorithms and their Applications, 1985, pp. 160–168

Authorized licensed use limited to: UNIVERSITAS GADJAH MADA. Downloaded on March 23,2024 at 07:22:09 UTC from IEEE Xplore. Restrictions apply.