Journal of Interdisciplinary Mathematics
ISSN: 0972-0502 (Print), ISSN: 2169-012X (Online)
Vol. 28 (2025), No. 1, pp. 95–105
DOI : 10.47974/JIM-1800

Development of a new approach algorithm for ant colony optimization using Python program

Hanan Abdul Hamza Hasan *
Ahmed Sabah Ahmed †
Department of Mathematics
University of Babylon
Babylon 51002
Iraq

Abstract

The world faces many challenges in our daily life, and the challenges as considered a mathematical model in a wide range and require different improvement methods and approaches. In this research, we reviewed one of the most important algorithms in swarm intelligence, which is the ant algorithm. One of its shortcomings is that it does not converge quickly, it is slow, and one of its advantages is that it is strong and capable of improvement. The research study adopts a new approach to develop ant algorithm to obtain the solution. The research worked on improving the algorithm with convergence speed and high search ability while ensuring the optimal solution. We applied the Ant algorithm to the street vendor. Optimization with high accuracy while reducing time.

Subject Classification (2020): *65K10, 49Mxx, 49M05.*

Keywords: *Particle swarm optimization, Numerical optimization, Design algorithms, Ant.*

1. Introduction

The natural world has frequently been a significant wellspring of inspiration for addressing intricate challenges across diverse domains of scientific inquiry and technological innovation. Ant algorithms are a notable

---

This article has been corrected with minor changes. These changes do not impact the academic content of the article.

* *ORCID-ID:* `0000-0001-5759-6413`
† *ORCID-ID:* `0000-0002-9033-4498`

* *E-mail:* `hanan1151983@gmail.com (Corresponding Author)`
† *E-mail:* `aljelawy2000@yahoo.com`

© Taru Publications

paradigm within the broad range of bio-inspired optimization approaches [1]. They offer solutions for intricate optimization and routing problems. The way that actual ants forage has an impact on the algorithms [2]. Ant algorithms, which fall under the category of swarm intelligence approaches, are based on the remarkable foraging abilities exhibited by ants these insect's exhibit collective intelligence through their collaborative behavior among colonies, despite their individual simplicity [3]. Ants have a navigational strategy wherein establish and adhere to pheromone trails to be efficiently locate the most favorable paths connecting their nests and sources of sustenance [4]. The aforementioned mode of communication, which was based on the principles of the stigmergy, has served as a source of inspiration for a novel set of optimization strategies that aim to replicate these innate processes. The captivating domain of ant algorithms [5]. In this study, we investigate the fundamental principles, mechanisms, and practical implementations associated with the subject matter. By elucidating the manner in which ant algorithms emulate the astute conduct of ants the capacity to address intricate predicaments spanning several fields, encompassing but not limited to the Traveling Salesman Problem (TSP), network routing, scheduling, and other related areas [6].The ant algorithms further, we will uncover the innovative methods via which these algorithms use collaboration and flexibility, thereby facilitating the development of efficient solutions for practical problems. The application of this technology spans across several industries such as transportation, logistics, and telecommunications, among others, so positioning it as a vital resource in the contemporary period focused on enhancing optimization and efficiency [7, 14-22].

Particle Swarm Optimization

James Kennedy, a psychologist and sociologist, and Russell Eberhart, an electrical engineer, discovered the bird swarm method (also referred to as particle swarm optimization, or PSO) in 1995, how these particles aggregate and behave certain ways to obtain food. An example is the ant algorithm and how it works in programming language [8].

Definition: Optimal solution

The optimal solution is the feasible solution that yields the highest attainable objective function value [9].

Definition: A feasible solution

is one that satisfies all constraints or a practical option that yields the best possible value of the objective function [10].

Optimization Problem

The optimization problem's overall basic formula is as follows

$$Minimize\begin{cases} f(z) \\ gi(z) \leq 0, \forall i \in \{1, \dots, m\} \\ hi(z) = 0, \forall i \in \{1, \dots, p\} \\ z \in R^n \end{cases}$$

$f: R^n \to R$. The objective function, often known as the cost function, is a real-valued function that we must reduce. The vector x represents a set of n independent variables [11]. $z = [z_1, z_2, \dots z_n]^T \in R^n$ Variables$z_1, z_2, \dots z_n$ are often referred to is Variables the decision-making. The set R^n subset ofR^n referred to as the practicable set or constraint set. The aforementioned optimization problem may be cone pluralized as a decision problem encompassing determining the choice variables' optimal vector z from all feasible vectors inR^n; by optimal, we mean the vector that yields the objective function's lowest value. This vector is referred to as the minimizer off over. There could be a large number of minimizers. It is sufficient to locate any of the minimizers. Additionally, there are optimization issues that need objective function to maximize. In that situation [23-26], we are in search of maximizers. Maximizers and minimizers consist of sometimes known as extremist difficulties, it equivalently represented is aforementioned minimization form, as maximizing as equivalent to minimizing the negation of the objective function (-f) consequently, we can focus just on minimization issues without sacrificing generality [12].

Karush-Kuhu-Tucker (KKT) Conditions: [13].

The following four conditions are referred KKT conditions (for a problem with Differentiable (f_i, h_i)

1. Primal constraints:$f_j(x) \leq 0, \mathrm{i} = 1, \dots \mathrm{I}, g_i(x) = 0, j = 1, \dots, E$
2. Complementary slackness:$\beta_j f_j(x) = 0, \mathrm{i} = 1, \dots \mathrm{I}$
3. Dual constraints:$\beta \geq 0$
4. Gradient of Lagrange with x vanishes (Figure 1):

$$\nabla f_o(x) + \sum_{i=1}^{I} B_i \nabla f_i(x) + \sum_{i,=1}^{E} Z_i \nabla g_i(x) = 0$$

The Ant Colony Optimization (ACO) algorithm consists of several key components

ACO is an optimization technique inspired by nature. As particularly well suited for combinatorial optimization issues as solved. Behavior demonstrated ant, the algorithm employs a probabilistic approach to detect solutions that are optimal or very close to optimal the algorithm known as Ant Colony Optimization (ACO) commonly utilized to address the Traveling Salesman Problem (TSP). Which entails the responsibility of ascertaining the optimal route for a sales representative to traverse a specified collection of cities while ensuring that each city is visited exclusively once.

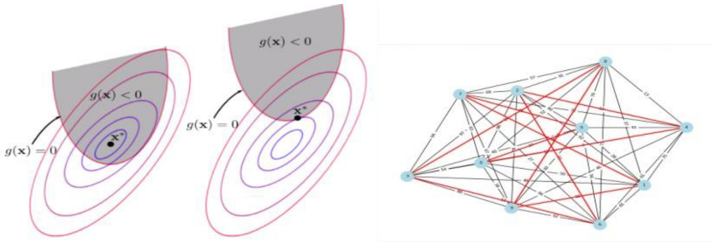

Figure 1
Graphical Explanation for the KKT Conditions

1. The preliminary stage of the procedure as denoted as initialization: Initializing the Hormone Matrix involves constructing a matrix. T (i, j) hitch signifies the concentration of pheromones what shows the amount of pheromones on the edge of the city i and city j hitch signifies the concentration of pheromones typically, it is common to assign an initial value to all entries. Such as 0.1

2. The steps used to come up with a solution**:** For every ant, a solution created, designated as. Begin from an initial urban center, either at random or by adopting a heuristic technique.

When there are cities that have not yet visited the next city as picked using a probability distribution that considers pheromone levels. (τ) In addition to a heuristic value (η)) that denotes the appeal of moving from one city to another.

3. Pheromone Update: Revise Upon completion of the construction of solutions by all ants, it is necessary to update the pheromone levels on the edges. The pheromone update can implemented by applying the following rule. Δτ(i, j) = Σ [Q / Lk] for all ants k that use the edge (i, j) Update τ(i, j) = (1 - ρ) * τ(i, j) + Δτ(i, j), ρ evaporation rate and Q is a never changes.

4. Finishing is Requirement: The process of constructing and updating should as repeated for a predetermined number of iterations or until a termination condition is satisfied

5. Outcome: The algorithm's result is the best answer that the ants were able to find.

Algorithm: The Ant Colony Optimization

Initialization

1. Initialization Pheromone Matrix:T(i, j) ← small constant
2. Define Parameters:$Q, \alpha, \beta,$ Iterative Process (for each iteration):
3. Ant Movement:.$p_{ij} = \frac{T_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{k \in allowedCities} T_{ik}^{\alpha}\eta_{ik}^{\beta}}$
1. Pheromone Update:.$T_{ij} \leftarrow (1-\rho), \quad \Delta T_{ij} = \frac{Q}{Total\ Length\ of\ the\ tour}.T_{ij} \leftarrow T_{ij} + \Delta T_{ij}$
2. Termination: Final Solution: Best Solution: Highest Pheromone tour

The Table provided shows the results of the Ant Colony Optimization (ACO) algorithm as it progresses through iterations (Table 1).

Table 1

The results of the Ant Colony Optimization (ACO)

Iteration	Shortest Path Length	Best Path	Pheromone Levels
1	295	[1, 9, 8, 7, 5,0,4,2,6,3]	[0.10694, 0.10694, 0.10694,...]
2	284	[8, 4, 2, 6, 3,1,7,5,0,9]	[0.11385, 0.11385, 0.11385, …]
3	284	[8, 4, 2, 6, 3, 1,7,5,0,9]	[0.12007, 0.12007, 0.12007,...]
4	284	[8, 4, 2, 6, 3, 1,7,5,0,9]	[0.12567, 0.12567, 0.12567,...]
5	284	[8, 4, 2, 6, 3,1,7,5,0,9]	[0.13071, 0.13071, 0.13071,...]

Let us break down the components of the table and explain how the ants are navigating and how each iteration affects the next:

1. **Iteration**: This column represents the current iteration number, from one to five in the example table
2. **Shortest Path Length**: This column displays the length of the shortest path found by the ACO algorithm in the given iteration. The goal of the algorithm is to minimize this path length

3. **Best Path**: This column shows the sequence of nodes that make up the shortest path found in the current iteration. For example, **[0, 7, 5, 8, 3, 2, 4, 9, 6, 1, 0]** represents a path starting and ending at node 0, passing through the other nodes in the given order
4. **Pheromone Levels**: This column displays the pheromone levels on the edges of the graph. Pheromone is a chemical substance deposited on the edges by the ants. It influences the ants' decision-making. Pheromone levels are shown in the format **[0.10000, 0.10000,]** for each edge, with up to 5 decimal places

Now, let us explain how the ACO algorithm works and how it progresses through iterations:

1. **Initialization**: In the first iteration, the algorithm initializes with a certain number of ants (in this case, 5), a graph representing the problem, and initial pheromone levels on the edges
2. **Ant Movement**: In each iteration, every ant starts from a random node. They choose the next node to visit based on a combination of pheromone levels and the distance to neighboring nodes. Ants apply a probabilistic rule to make these decisions
3. **Path Building**: As ants move, they build paths by sequentially selecting nodes and updating the path's total length
4. **Pheromone Update**: After each ant completes its tour, the algorithm updates the pheromone levels on the edges. Edges that are part of shorter paths receive more pheromone, while longer paths receive less. This process involves both pheromone evaporation (a decrease in pheromone levels) and pheromone deposition (an increase in pheromone levels) on selected edges
5. **Best Path Update**: The algorithm keeps track of the best path found so far and the corresponding length. If an ant discovers a shorter path in the current iteration, it updates the best path and length
6. **Iteration Progression**: The algorithm repeats these steps for the specified number of iterations. In each iteration, the ants' exploration of the problem space and the pheromone update process influence their decisions in the next iteration. Over time, shorter paths are more likely to be discovered and receive higher pheromone levels

7. **Result**: After all iterations are complete, the algorithm presents the result, which is the best path found overall, along with its length

The ACO algorithm converges to a solution by iteratively improving the paths taken by the ants, as shorter paths accumulate more pheromone. This allows it to find a good solution to optimization problems by mimicking the foraging behavior of ants. The algorithm balances exploration (discovering new paths) and exploitation (reinforcing good paths) to reach an optimal or near-optimal solution.

Final Result: Shortest Path: [2, 4, 0, 5, 7, 1, 9, 8, 3, 6]

Shortest Path Length: 220 Figure 1: the

Final Result Explanation:

- The ACO algorithm has successfully found the shortest possible route for the traveling salesman to visit all locations and return to the starting point. This solution minimizes the total distance traveled, making it an efficient and cost-effective path
- The graph representation and drawing provide a visual confirmation of the optimized path and the quality of the solution.
- The red-highlighted path within the drawing is the actual route that the traveling salesman should follow to minimize travel distance
- The associated numerical value for the "Shortest Path Length" quantifies the total distance of the optimized route.

In summary, the result and drawing demonstrate that the ACO algorithm has effectively solved the TSP for the given dataset, providing a route that minimizes travel distance. This graphical representation makes it easy to understand and communicate the quality of the solution, making it a valuable tool for decision-making in logistics and route optimization

All subsequent test cases adhere to the same underlying methodology. The fundamental approach remains consistent across all test cases, with variations introduced only in terms of the number of cities and the specific 2D coordinates used. (for example see Figure 2).

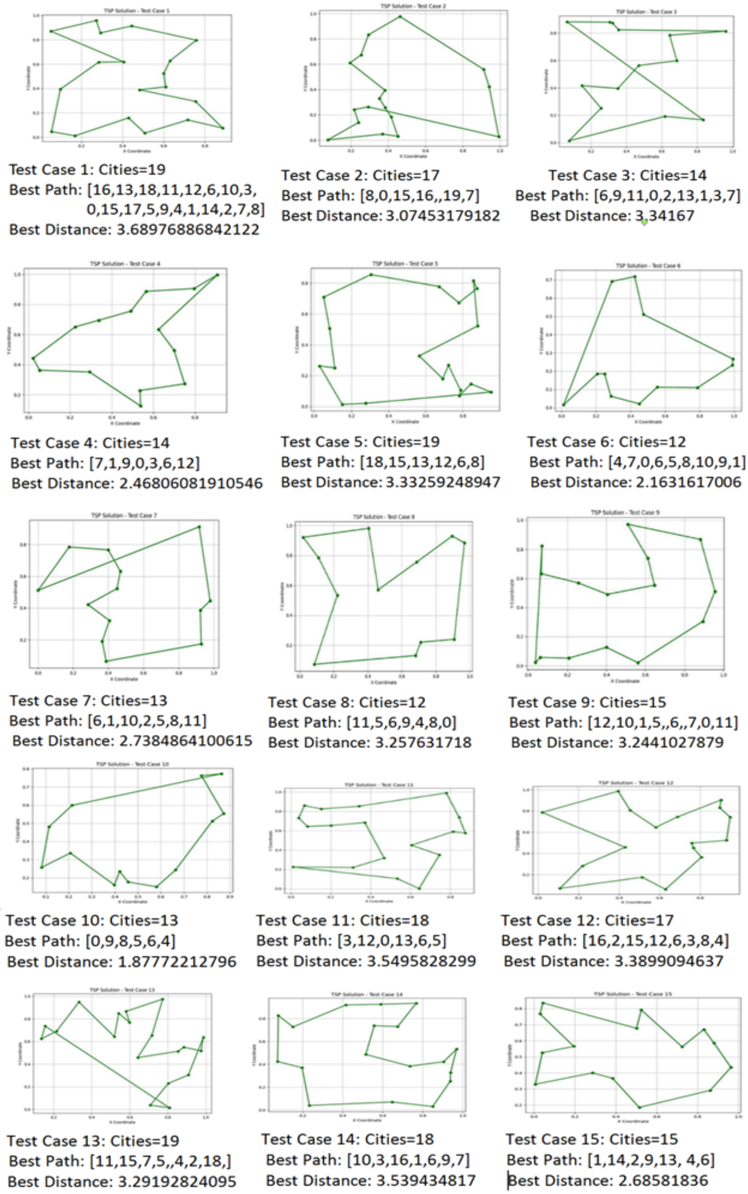

Figure 2

Test Case 1: Cities=19 Best Path: [16, 13, 18, 11, 12, 6, 10, 3, 0, 15, 17, 5, 9, 4, 1, 14, 2, 7, 8].

This ensures a rigorous and uniform assessment of the Ant Colony Optimization (ACO) algorithm's performance across a range of scenarios, maintaining the integrity of the experimental design (see **Figures 2**).

Conclusion

Based on the results that we obtained by conducting the test using the Python language, the new approach was adopted in that it relies on the ants tracking the best path along the corresponding length and is more accurate and flexible in the ants adopting the best paths, taking into account the ants' behavior in choosing food with great taste, as the results demonstrated the discovery of new paths to reach the closest optimal solution

References

[1] B. Shuang, J. Chen, and Z. Li, "Study on hybrid PS-ACO algorithm," *Applied Intelligence*, vol. 34, no. 1, pp. 64–73 (2011).

[2] S. Li, Y. Wei, X. Liu, H. Zhu, and Z. Yu, "A new fast ant colony optimization algorithm: the saltatory evolution ant colony optimization algorithm," *Mathematics*, vol. 10, no. 6, p. 925 (2022).

[3] X. Xie, Z. Tang, and J. Cai, "The multi-objective inspection path-planning in radioactive environment based on an improved ant colony optimization algorithm," *Progress in Nuclear Energy*, vol. 144, p. 104076 (2022).

[4] L. Wu, X. Huang, J. Cui, C. Liu, and W. Xiao, "Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot," *Expert Systems with Applications*, vol. 215, p. 11941 (2023).

[5] S. Kashef and H. Nezamabadi-Pour, "An advanced ACO algorithm for feature subset selection," *Neurocomputing*, vol. 147, pp. 271–279 (2015).

[6] T. Liao, T. Stützle, M. A. M. de Oca, and M. Dorigo, "A unified ant colony optimization algorithm for continuous optimization," *European Journal of Operational Research*, vol. 234, no. 3, pp. 597–609 (2014).

[7] X. Chen, Y. Kong, X. Fang, and Q. Wu, "A fast two-stage ACO algorithm for robotic path planning," *Neural Computing and Applications*, vol. 22, pp. 313–319 (2013).

[8] A. G. Gad, “Particle swarm optimization algorithm and its applications: a systematic review,” *Archives of Computational Methods in Engineering,* vol. 29, no. 5, pp. 2531–2561 (2022).

[9] R. Dorfman, P. A. Samuelson, and R. M. Solow, *Linear Programming and Economic Analysis.* Courier Corporation (1987).

[10] G. B. Dantzig, Linear Programming and Extensions. Princeton University Press (1998).

[11] X.-S. Yang, Optimization and Metaheuristic Algorithms in Engineering, Metaheuristics in Water, Geotechnical, and Transport Engineering, vol. 1, p. 23 (2013).

[12] D. P. Bertsekas, “Nonlinear programming,” *Journal of the Operational Research Society,* vol. 48, no. 3, p. 334 (1997).

[13] R. C. Robinson, Introduction to Mathematical Optimization, Department of Mathematics, Northwestern University, Illinois, USA (2013).

[14] M. R. Farahani, S. Jafari, S. A. Mohiuddine, and M. Cancan, “Intuitionistic fuzzy stability of generalized additive set-valued functional equation via fixed point method,” *Mathematical Statistician and Engineering Applications,* vol. 71, no. 3s3, pp. 142–154 (2022). [Online]. Available: https://www.philstat.org.ph/index.php/MSEA/article/view/355.

[15] M. R. Farahani, J. E. N. Valdes, M. Cancan, S. Jafari, and C. Park, “Fuzzy stability in the sense of Hyers-Ulam-Rassias of a functional equation on quadratic forms,” *J. Interdiscip. Math.,* vol. 26, no. 6, pp. 1355–1365. [Online]. Available: https://doi.org/10.47974/JIM-1633.

[16] S. M. Kadham and M. A. Mustafa, “Medical applications of the new-transform,” *J. Interdiscip. Math.*, vol. 26, no. 6, pp. 1341–1353. [Online]. Available: https://doi.org/10.47974/JIM-1632.

[17] S. M. Kadham and M. A. Mustafa, “Fuzzy SHmath. Mbio-transform generalization and application to skin cancer imaging (distributed diseases),” *J. Interdiscip. Math.*, vol. 26, no. 6, pp. 1031–1042. [Online]. Available: https://doi.org/10.47974/JIM-1603.

[18] R. H. Hasan, M. A. Kadhim, and A. M. Darweesh, “Fuzzy partial T-transforms applied to a system of PDE,” *J. Interdiscip. Math.*, vol. 26, no. 6, pp. 1099–1107. [Online]. Available: https://doi.org/10.47974/JIM-1609.

[19] S. Jafari, M. R. Farahani, and M. Rajesh, "Double fuzzy rarely continuous functions," *Annals of Fuzzy Mathematics and Informatics*, vol. 26, no. 1, pp. 35–45.

[20] M. B. Salman and W. H. Hanoon, "Fuzzy scl-semi 2-ABSO submd and related concepts," *J. Interdiscip. Math.*, vol. 26, no. 6, pp. 1231–1242. [Online]. Available: https://doi.org/10.47974/JIM-1621.

[21] A. S. Abed and A. T. Hameed, "The fuzzy results on SA-algebra," *J. Interdiscip. Math.*, vol. 26, no. 6, pp. 1273–1280. [Online]. Available: https://doi.org/10.47974/JIM-1625.

[23] Z. H. Abdulabbas and L. A. R. Al Asadi, "Calculating Dynamic Strengths of Concrete Subjected to Impact Load," *Key Engineering Materials,* vol. 895, pp. 12–19 (Aug. 2021), doi: https://doi.org/10.4028/www.scientific.net/kem.895.12.

[24] R. A. Kamil, H. M. Atiea, A. A. Kadhem, Layth Abdulrasool Alasadi, and Q. A. Jabal, "Improvement of the mechanical properties of polymer-modified concrete by using PC superplasticizer," Pollack Periodica, vol. 18, no. 3, pp. 14–19 (Jul. 2023), doi: https://doi.org/10.1556/606.2023.00742.

[25] S. M. Kadham, "Acute interstitial pneumonia image enhancement using fuzzy partial transforms," *Applied Geomatics,* vol. 16, no. 1, pp. 35–39 (May 2023), doi: https://doi.org/10.1007/s12518-023-00509-8.

[26] H. S. Al-Bahrani, L. A. Al-Asadi, L. K. Al-Waeli, H. F. Al-Isawi, and A. M. Al-Jobory, "Determining new way to detect hydrocarbon polluted water at vertical and horizontal spaces," *Materials Today: Proceedings,* vol. 42, pp. 2071–2075, 2021, doi: https://doi.org/10.10

Received June, 2024