Stallion Journal for Multidisciplinary Associated Research Studies
ISSN (Online): 2583-3340
Volume-5 Issue-4 || August 2026 || PP. 1-9
https://doi.org/10.55544/sjmars.5.4.1

On the Development and Application of a Modified Ant Colony Optimization (MACO) Algorithm for Solving Optimization Problems

Adebayo Kayode James[1*], Omowaye Solomon Kehinde[1], and Adetolaju Olu Sunday[2]
[1]Department of Mathematics, Faculty of Physical Sciences, Ekiti State University, Ado Ekiti, NIGERIA
[2]Department of Computer Science, Faculty of Physical Sciences, Ekiti State University, Ado Ekiti, NIGERIA

[*]Corresponding Author: kayode.adebayo@eksu.edu.ng

ORCiD

https://orcid.org/0000-0003-3684-6190

www.sjmars.com || Vol. 5 No. 4 (2026): August Issue

Date of Submission: 16-07-2026	**Date of Acceptance:** 21-07-2026	**Date of Publication:** 03-08-2026

ABSTRACT

This study presents the development and application of a Modified Ant Colony Optimization (MACO) algorithm incorporating constant variables to enhance solution efficiency in optimization problems. In the traditional Ant Colony Optimization (ACO) algorithms, though effective in solving combinatorial problems, they often suffer from slow convergence and premature stagnation due to excessive dependence on pheromone and parameter stochastic update settings. The MACO algorithm introduces constant variables control that regulates pheromone evaporation rate, heuristic influence, and exploration–exploitation balance, thereby improving algorithm stability and convergence speed. Additionally, the modification integrates features from complementary metaheuristics to strengthen global search capability while maintaining local refinement efficiency. The model is formulated and tested on benchmark optimization problems, demonstrating improved performance in term of the solution quality, computational rigor, time, and robustness compared to traditional ACO approaches. Results indicate that the incorporation of constant variables reduces parameter uncertainty and enhances repeatability of outcomes. The study contributes to the development of swarm intelligence techniques by providing a more controlled, flexible, and reliable optimization structure suitable for applications in engineering, routing, and logistics design problems.

Keywords- Modified Ant Colony Optimization (MACO) Algorithm, Optimization Problems, Ant Colony Optimization (ACO), Max–Min Ant System (MMAS).

I. INTRODUCTION

Optimization problems remain central to operations research, engineering, and applied mathematics, particularly in systems requiring efficient decision-making under constraints. Classical optimization techniques often struggle with nonlinear, high-dimensional, and combinatorial problems, leading to the adoption of meta-heuristic techniques that presents near-optimal solutions within reasonable time frame. Among these, the Ant Colony Optimization (ACO) technique developed by Dorigo and his colleagues, has gained ground due to its biologically inclined mechanism that based on the foraging behavior of ants and by extension communication via pheromone trails. ACO has been applied to problems such as the Traveling Salesman Problems (TSP), [1] and [2] Vehicle Routing Problems, and scheduling demonstrating efficiency, flexibility, and robustness in diverse domains.

Despite its success, the standard ACO algorithm is not without limitations. One major challenge is premature convergence, where ants tend to follow strongly reinforced paths too early, reducing exploration of the search space.

Additionally, the stochastic nature of ACO parameters, such as pheromone evaporation rate and heuristic influence, can lead to inconsistent performance and sensitivity to parameter tuning. These issues often bring about zigzagging convergence profile and suboptimal solutions, particularly in large-scale or dynamic environments. Researchers have therefore explored various modifications, including adaptive parameter tuning and hybridization strategies, to address these shortcomings [3] and [4].

Modified Ant Colony Optimization (MACO) algorithms have emerged as an effective approach to improving the performance of classical ACO. By fussing ACO technique together with other optimization techniques such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), or local search heuristics, MACO leverages the strengths of multiple methods to achieve better global and local search capabilities. The modification enhances diversification and intensification processes, that brings about improved convergence behavior and solution quality. Studies have shown that MACO algorithms outperform standalone ACO in solving complex combinatorial optimization problems, including routing and network design [5] and [6].

In recent developments, the introduction of constant variables into MACO structures has attracted attention as a means of stabilizing algorithm performance. Constant variables serve as fixed control parameters that regulate key processes such as pheromone updating, heuristic weighting, and exploration–exploitation balance. Unlike adaptive or random parameters, constant variables reduce uncertainty and improve reproducibility of results. This approach is particularly useful in applications where consistent performance is required across multiple runs, such as logistics planning and engineering optimization. The use of constant parameters also simplifies algorithm design and reduces computational overhead associated with parameter tuning [7] and [8].

Furthermore, incorporating constant variables into MACO contributes to a more structured and controlled search process. By fixing certain parameters, the algorithm can maintain a steady state between exploration of the solutions of known existing solutions. This reduces the likelihood of stagnation and enhances convergence speed without sacrificing solution diversity. In addition, constant-variable MACO models can be easily extended to solve real-world optimization problems, including disrupted vehicle routing problems (DVRP), where system uncertainties and dynamic changes require robust and stable optimization techniques [9] and [10].

II. LITERATURE REVIEW

The Ant System (AS), introduced by Dorigo, *et al*., represents the foundational architectural structure of the Ant Colony Optimization and has been studied widely for solving problems in combinatorial optimization such as TSP. In AS, as reported in [11], synthetic ants conceive solutions probabilistically subject to pheromone trails and heuristic information. The transition probability of an ant that moved from node (i, j) is given by:

$$P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{i\in N_i^k}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}} \quad (1)$$

where the pheromone intensity is represented τ_{ij}, η_{ij} is the heuristic desirability, and α, β control their influence. The pheromone update rule in AS is expressed as:

$$\tau_{ij} \longleftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \quad (2)$$

where ρ is the rate of evaporation and τ_{ij}^{k} is the quantity of pheromone deposited by ant k. Although AS demonstrates strong exploration capability, it suffers from slow convergence and susceptibility to premature stagnation due to uncontrolled pheromone accumulation [1] and [10].

To tackle the limitations on the basic AS, the ACO structure was extended with improved pheromone updating strategies and heuristic control mechanisms. ACO generalizes the AS approach by incorporating adaptive search behaviors and refined exploitation strategies. One of the widely adopted formulations includes the global pheromone update rule:

$$\tau_{ij} \longleftarrow (1-\rho)\tau_{ij} + \rho\sum_{k=1}^{m}\Delta\tau_{ij}^{best} \quad (3)$$

where only the best-performing ant contributes to pheromone reinforcement. This modification enhances convergence speed while maintaining solution quality. ACO algorithms have been successfully applied in routing, scheduling, and network optimization, but they still exhibit sensitivity to parameter settings, particularly in large-scale problems [2] and [10].

The Max–Min Ant System (MMAS) was introduced as a significant improvement over standard ACO by imposing explicit limits on pheromone trail values to prevent premature convergence. In MMAS, pheromone values are restricted within the pre-defined bounds:

$$\tau_{min} \leq \tau_{ij} \leq \tau_{max} \quad (4)$$

Additionally, only the best ant (iteration-best or global-best) is allowed to update pheromones, ensuring controlled exploitation of promising regions in the search space. The pheromone update rule in MMAS is expressed as:

$$\tau_{ij} \longleftarrow (1-\rho)\tau_{ij} + \Delta\tau_{ij}^{best} \quad (5)$$

Stallion Journal for Multidisciplinary Associated Research Studies
ISSN (Online): 2583-3340
Volume-5 Issue-4 || August 2026 || PP. 1-9
https://doi.org/10.55544/sjmars.5.4.1

This approach significantly reduces stagnation and improves convergence reliability. However, the performance of MMAS is still dependent on careful parameter tuning, particularly the selection of τ_{min} and τ_{max}, which can vary across problem instances [13], [5], [14], and [4].

Recent research has focused on hybridizing these ACO variants to develop Modified Ant Colony Optimization (MACO) algorithms that combine the strengths of AS, ACO, and MMAS. MACO structures integrate multiple pheromone update strategies, local search procedures, and complementary metaheuristics such as Genetic Algorithms (GA) or Particle Swarm Optimization (PSO). The hybridization enhances both exploration and exploitation by dynamically balancing global search with local refinement. In such models, constant variables are increasingly introduced to stabilize algorithm performance by fixing key parameters such as α, β *and* ρ thereby reducing variability across runs and improving reproducibility and [12].

The incorporation of constant variables into MACO algorithms represents a structured approach to parameter control. Unlike adaptive or stochastic parameter tuning methods, constant-variable MACO maintains fixed values for critical parameters, ensuring consistent behavior throughout the optimization process. For instance, fixing the evaporation rate ρ and heuristic weight β allows the algorithm to maintain a stable balance between exploration and exploitation. This is particularly beneficial in solving deterministic optimization problems where repeatability is essential. Studies have shown that constant parameterization can reduce computational complexity and improve convergence speed without significantly compromising solution quality [3] and [1].

Furthermore, the integration of AS, ACO, and MMAS within a MACO structure allows for a more robust optimization strategy. The AS component contributes to initial exploration, ACO enhances adaptive learning through pheromone reinforcement, and MMAS ensures controlled convergence through bounded pheromone trails. When combined with constant variables, the hybrid model achieves improved stability and reduced sensitivity to parameter fluctuations. This approach has proven effective in solving complex real-world problems such as vehicle routing, network design, and scheduling under uncertainty [14] and [8] In summary, the literature indicates that the evolution from AS to ACO and MMAS has significantly improved the performance of ant-based optimization algorithms. The development of MACO algorithms incorporating constant variables represents a further advancement, offering enhanced convergence, stability, and computational efficiency. By combining the strengths of multiple ACO variants and introducing controlled parameterization, MACO provides a powerful structure for addressing complex optimization challenges in modern applications [2] and [7].

III. METHODOLOGY

The Modified Ant Colony Optimization (MACO) algorithm is such that is described in the steps below:

Step 1: Parameters are initialized: the number of ants is represented by N, $x_{i,j}$ is the value assigned to each path j at iteration, i. Where n is the number of paths, ρ denotes the rate at which pheromone evaporates, and τ_0 is the amount of the initial pheromone deposit. Keeping $\boldsymbol{i}$ at $\mathbf{0}$ and the predetermined tolerance at $\boldsymbol{\varepsilon} \leq \mathbf{0.00001}$.

Step 2: Evaluate the Transition Probability, when $i = 0$ as

$$TP_{i,j} = \frac{\tau_0}{n} \quad (6)$$

With (6) compute the Cumulative Transition Probability, $(CTP_{i,j})$, using the relation:

$$TP_{1,\mathrm{j}} = \frac{M_{i-1}}{(n-freq(f_{best}))(M_{i-1}) + \tau_1} \quad (7)$$

The expression (7) applies to $i \geq 1$, while $i = 0$ applies to the Transition Probability, $TP_{1,\mathrm{j}}$, and the Jump Transition Probability, $TP_{i,j}^{jump}$ will be quantified using

$$TP_{i,j}^{jump} = \frac{\tau_i}{(n-freq\ of\ f_{best})(M_{i-1}) + \tau_i} \quad (8)$$

in place of (7) where $freq. of\ f_{best}$ denotes the function highest value frequency. Both the $TP_{i,j}$ and the $TP_{i,j}^{jump}$ are employed to evaluate the Cumulative Transition Probability, $(CTP_{i,j})$. At iteration $i \geq 1$ and path j, the $CTP_{i,j}$, is given as:

$$\left.\begin{aligned}
CTP_{i,1} &= TP_{i,1} \\
CTP_{i,2} &= CTP_{i,1} + TP_{i,1} \\
&\vdots \\
CTP_{i,m} &= CTP_{i,m-1} + TP_{i,1} \\
\boldsymbol{CTP_{i,m+1}} &= \boldsymbol{CTP_{i,m}} + \boldsymbol{TP_{i,j}^{jump}} \\
CTP_{i,m+2} &= CTP_{i,m+1} + TP_{i,1} \\
CTP_{i,m+3} &= CTP_{i,m+2} + TP_{i,1} \\
&\vdots \\
CTP_{i,n} &= CTP_{i,(n-1)} + TP_{i,1}
\end{aligned}\right\} \quad (9)$$

Stallion Journal for Multidisciplinary Associated Research Studies
ISSN (Online): 2583-3340
Volume-5 Issue-4 || August 2026 || PP. 1-9
https://doi.org/10.55544/sjmars.5.4.1

The transition probability *jump* usually would occurs after the first iteration at a point when the $CTP_{i,j}$ has experience an unprecedented rise in the value of function compared with other points that correspond to the best and at least one paths will have a higher pheromone concentration that implies that more ants have traversed the route [11].

At the point when $i = 0$ in (9) all the paths is assumed to have equi-probable chances of being traversed by the ants. Then, such path is acclaimed the best path because since it is being traversed by most of the ants against when it is compared to other paths. As a result, the transition probability jump (i.e. multiple ants taking this path) is regarded as the best path. Here, m distinguishes the point at which the transition probability jump occurs from the other paths traversed by the ants.

Step 3: (a) Generate N random constant numbers, $r_{1,} r_2, \dots, r_N$ corresponding to the number of ants within the closed range of 0 to 1,

(b) Correspond each of the randomized constant numbers, r_N, with the equivalent $CTP_{i,j}$, or approximated value to $CTP_{i,j}$, using the relation:

$r_N \cong CTP_{i,j}$ at path j of $x_{i,j}$ (10)

After the first iteration where the randomized constant numbers are generated, the randomized constant numbers are kept constant over subsequent iteration.

Step 4: (a) **Compute the objective function value for each ant, $f(x_{i,j})$.**

(b) Determine the best and worst path(s) from the values objective function.

In a case of a maximization problem, the best path, f_{best}, is the highest objective function value

f_{best} is the Highest of $\left(f(x_{i,1}), f(x_{i,2}), \dots, f(x_{i,N})\right)$ (11)

while the worst path, f_{worst}, is the least objective function value

f_{worst} is the Lowest of $\left(f(x_{i,1}), f(x_{i,2}), \dots, f(x_{i,N})\right)$ (12)

Step 5: **Test for convergence.**

The convergence of the entire process can be determined by either or all of the following:

(1) if the objective function values are all the same or approximately the same, the system is said to have converged i.e.

$f(x_{i,1}) = f(x_{i,2}) = \cdots = f(x_{i,N})$ or (13)

(2) determine if the absolute value of two consecutive function values is less or equal to a predetermined constant, ε as

$|f(x_{i,N}) - f(x_{i,N-1})| \leq \varepsilon$ (14)

When any of the convergence criteria is met then, terminate the process. Else, progress to step 6.

Step 6: (a) Calculate the Pheromone Remnant, M_i, with:

$M_i = (\sum_{i=1}^{N} \tau_i) - \rho$. (15)

where ρ is the pheromone evaporation rate at iteration $i = 0$ that is predefined by the user.

(b) Compute the objective function ratio value given by

$\Delta\tau_{j,k} = \frac{\xi \sum freq\ of\ f_{best}(f_{best})}{f_{worst}}$ (16)

$j = 1, \dots, n$ and $k = 1, \dots, N$

(c) Update the pheromone deposit as follows:

$\tau_{i+1,j} = M_{i-1} + \sum_{k=1}^{N} \Delta\tau_{i,j}$ (17)

where M_{i-1} in (17) is regarded as the pheromone remnant and the expression on the right hand side of (17) is the operator that determines the pheromone deposit that might be equal to zero, by [11] and [15] if none of the ant passes the node (i, j) at the last iteration. Here, i, j, and $k = 1, 2, \dots, N$ is the iteration counter, the path counter, and stands for each of the ants respectively. Then, go back to Step 2.

Test Problem 1:

Suppose a manufacturing industry with the following data

Price = x,

Unit sales = $70000 - 200x$

Selling Price = Unit Sales × Price = $(70000 - 200x) \times x = 70000x - 200x^2$

Cost Price = $8400000 - 22000x$

Profit = Selling Price - Cost Price = $-200x^2 + 92000x - 8400000$

The profit function is therefore given by:

$Q(x) = -200x^2 + 92000x - 8400000$

Subject to $220 \leq x \leq 300$

Hence, one is to calculate the quantity to be produced and at best price will is be at its best?

i.e. Maximize Q$(x) = -200p^2 + 92000p - 840000$

Subject to $220 \leq x \leq 300$)

Ref [12] and [16].

Stallion Journal for Multidisciplinary Associated Research Studies
ISSN (Online): 2583-3340
Volume-5 Issue-4 || August 2026 || PP. 1-9 https://doi.org/10.55544/sjmars.5.4.1

$f(x) = -200x^2 + 92000x - 8400000$ (18)
Solution:
Step 1: Number of Ants = 4, $n = 9, \rho = 0.5, \tau_0 = 1, \zeta = 2$
$x_1, = 220\ , x_2 = 230, x_3 = 240, x_4 = 250, x_5 = 260, x_6 = 270, x_7 = 280, x_8 = 290, x_9 = 300$
Step 2: $TP_{i,j} = \frac{\tau_0}{n} = \frac{1}{9} = 0.1111111111111$ (19)
$CTP_{1,1} = TP_{1,1} = 0.1111111111$
$CTP_{1,2} = CTP_{1,1} + TP_{1,1} = 0.1111111111111 + 0.11111111111 = 0.2222222222$
$CTP_{1,3} = CTP_{1,1} + TP_{1,2} = 0.1111111111111 + 0.2222222222 = 0.33333333333$
$CTP_{1,4} = CTP_{1,1} + TP_{1,3} = 0.1111111111111 + 0.33333333333 = 0.4444444445$
$CTP_{1,5} = CTP_{1,1} + TP_{1,4} = 0.1111111111111 + 0.4444444445 = 0.55555555556$
$CTP_{1,6} = CTP_{1,1} + TP_{1,5} = 0.1111111111111 + 0.5555555556 = 0.666666666667$
$CTP_{1,7} = CTP_{1,1} + TP_{1,6} = 0.1111111111111 + 0.666666666667 = 0.77777777778$
$CTP_{1,8} = CTP_{1,1} + TP_{1,7} = 0.1111111111111 + 0.777777777778 = 0.8888888889$
$CTP_{1,9} = CTP_{1,1} + TP_{1,8} = 0.1111111111111 + 0.888888888889 = 1.0000000000$
N. B: Since no paths has been identified as f_{best} at the first iteration, the jump transition probability needs not be computed at this stage.
Step 3: (a). Generate N random numbers, r_1, r_2, r_3, and r_4 in the interval [0,1]
$r_1 = 0.3122, r_2 = 0.8701, r_3 = 0.4729, r_4 = 0.6190$
(b). $\boldsymbol{r_a \cong CTP_{i,j}}$ **at path $\boldsymbol{j}$ of $\boldsymbol{X_{i,j}}$** (20)
Correspond each of the randomized constant numbers, r_N, with the equivalent $CTP_{i,j}$, or approximated value to $CTP_{i,j}$, using the relation $r_N \cong CTP_{i,j}$ at path j of $x_{i,j}$
At the first iteration the randomized constant numbers are generated. The randomized constant numbers are then kept constant over subsequent iterations.
$r_1 = CTP_{13,} = x_3 = 240, r_2 = CTP_{1,8} = x_8, = 290, r_3 = CTP_{1,5} = x_5 = 260,\ r_4 = CTP_{1,6} = x_6 = 270$
Step 4: (a) $\boldsymbol{f(x_{i,j}) = f_a = CTP_{i,j}}$ (21)
$\boldsymbol{A}nt1: r_1 = 0.3122 \cong CTP_{1,3} = 0.33333333333\ then\ x_3 = 240, Hence\ f(x_1) = f(240) = 2160000$
$\boldsymbol{A}nt2: r_1 = 0.8701 \cong CTP_{1,8} = 0.8888888889\ then\ x_8 = 290, Hence\ f(x_8) = f(290) = 1460000$
$\boldsymbol{A}nt3: r_3 = 0.4729 \cong CTP_{1,5} = 0.55555555556\ then\ x_5 = 260,$
Hence, $f(x_5) = f(260) = 2000000$
$\boldsymbol{A}nt4: r_4 = 0.4729 \cong CTP_{1,6} = 0.6666666667\ then\ x_6 = 270,$
Hence $f(x_6) = f(270) = 1860000$
(b) Determine the $\boldsymbol{f_{best}}$ as well as the $\boldsymbol{f_{worst}}$
The f_{best} path = $f(x_1) = 2160000$, (22a)
The worst path = $f(x_2)$= 1460000 (22b)
Step 5: To find the Convergence
Step 6: (a) $\boldsymbol{M_i = \{\sum_{i=1}^{N} \tau_i\}\rho = \tau_0\ \rho} = 1 \times 0.5 = 0.5 = \boldsymbol{M_0}$ (23)
(b) $\Delta\boldsymbol{\tau_{i,k}} = \frac{\boldsymbol{\xi \sum freq\ of\ f_{best}\ (f_{best})}}{\boldsymbol{f_{worst}}} = \frac{2 \times 1 \times 2160000}{1460000} = 2.9589041095890$
(c) $\tau_1 = M_{i-1} + \sum_{k=1}^{N} \Delta\tau_{1,1} = 0.5 + 2.9589041095890 = 3.458\,9041095890$ (24)

Back to step 2
Step 2: $TP_{2,1} = \frac{M_{i-1}}{(n - freq\ of\ f_{best})(M_{i-1}) + \tau_i} = \frac{M_0}{(n - freq\ of\ f_{best})(M_0) + \tau_1}$ (25)
$TP_{2,1} = \frac{0.5}{(9-1)0.5 + 3.4589041095890} = 0.06703397612$
$TP_{2,3}^{jump} = \frac{\tau_i}{(n - freq\ of\ f_{best})(M_{i-1}) + \tau_i} = \frac{3.4589041095890}{7.4589041095890} = 0.42857142857$ (26)
$CTP_{2,1} = TP_{2,1} = 0.06703397612$
$CTP_{2,2} = CTP_{2,1} + TP_{2,1} = 0.13406795224$
$CTP_{2,3} = CTP_{2,2} + TP_{2,3}^{jump} = 0.13406795224 + 0.42857142857 = 0.56263938081$
$CTP_{2,4} = CTP_{2,3} + TP_{2,1} = 0.56263938081 + 0.06703397612 = 0.62967335693$
$CTP_{2,5} = CTP_{2,4} + TP_{2,1} = 0.6967073305\ , CTP_{2,6} = CTP_{2,5} + TP_{2,1} = 0.76374130917$
$CTP_{2,7} = CTP_{2,6} + TP_{2,1} = 0.83077528529, CTP_{2,8} = CTP_{2,7} + TP_{2,1} = 0.89780926141$
$CTP_{2,9} = CTP_{2,8} + TP_{2,1} = 0.96484323753$
Step 3: (a) Generate N random numbers r_1, r_2, r_3, and r_4 in the interval [0,1] without repeating random number.
$r_1 = 0.3122, r_2 = 0.8701, r_3 = 0.4729$, and $r_4 = 0.6190$ (27)

Stallion Journal for Multidisciplinary Associated Research Studies
ISSN (Online): 2583-3340
Volume-5 Issue-4 || August 2026 || PP. 1-9 https://doi.org/10.55544/sjmars.5.4.1

(b). $\boldsymbol{r_a \cong CTP_{i,j}}$ **at path** $\boldsymbol{j}$ **of** $\boldsymbol{X_{i,j}}$ (28)
$r_1 \equiv CTP_{2,3} \equiv x_3, r_2 \equiv CTP_{2,8} \equiv x_8, r_3 \equiv CTP_{2,3} \equiv x_{3,} r_4 \equiv CTP_{2,8} \equiv x_4$
Step 4: (a) $\boldsymbol{Ant\ i = f_a = f(x_i) = CTP_{i,j}}$ (29)

$\boldsymbol{Ant}1: r_1 = 0.3122 \cong CTP_{1,3} = 0.33333333333$ then $x_3 = 240$,
Hence $f(x_1) = f(240) = 2160000$
$\boldsymbol{Ant}2: r_2 = 0.8701 \cong CTP_{1,7} = 0.8888888889$ then $x_7 = 280$,
Hence $f(x_7) = f(280) = 1680000$
$\boldsymbol{Ant}3: r_3 = 0.4729 \cong CTP_{1,3} = 0.33333333333$ then $x_3 = 240$,
Hence $f(x_1) = f(240) = 2160000$
$\boldsymbol{Ant}4: r_4 = 0.6190 \cong CTP_{1,4} = 0.55555555556$ then $x_4 = 250$,
Hence, $f(x_4) = f(250) = 2100000$
(b) Determine the $\boldsymbol{f_{best}}$ and $\boldsymbol{f_{worst}}$
The $\boldsymbol{f_{best}} = f_1 = f_3 = f(x_3) = 2160000.$ (30a)
The $\boldsymbol{f_{worst}} = f_2 = f(x_7) = f(280) = 1680000$ (30b)
Step 6: (a) $\{\sum_{i=1}^{N} \tau_i\}\rho = \tau_0\, \rho = 0.5 \times 0.5 = 0.25 = M_1$ (31)
(b) $\Delta\tau_{i,k} = \frac{\xi \sum freq\ of\ f_{best}\ (f_{best})}{f_{worst}} = \frac{2 \times 2 \times 2160000}{1680000} = 5.14285714286$ (32)
$\Delta\tau_{2,1} = 5.14285714286$
(c) $\tau_2 = M_{i-1} + \sum_{k=1}^{N} \Delta\tau_{2,1} = 0.25 + 5.14285714286 = 5.39285714286$ (33)
Back to step 2
$TP_{3,1} = \frac{M_{i-1}}{(n-freq\ of\ f_{best})(M_{i-1})+\tau_i} = \frac{M_1}{(n-freq\ of\ f_{best})(M_0)+\tau_2}$ (34)
$TP_{3,1} = \frac{0.25}{(9-2)0.25+8} = \frac{0.25}{7 \times 0.25+5.39285714286} = 0.035$
$TP_{3,3}^{jump} = \frac{\tau_2}{(n-freq\ of\ f_{best})(M_1)+\tau_i} = \frac{5.39285714286}{7 \times 0.25+5.39285714286} = 0.7550000008$ (35)
$CTP_{3,1} = TP_{3,1} = 0.035$
$CTP_{3,2} = CTP_{3,1} + TP_{3,1} = 0.070$
$CTP_{3,3} = CTP_{3,2} + TP_{3,3}^{jump} = 0.070 + 0.7550000008 = 0.8250000008$ (36)
$CTP_{3,4} = 0.860000008$ $CTP_{3,5} = 0.8950000008$
$CTP_{3,6} = 0.9300000008$ $CTP_{3,7} = 0.9650000008$
$CTP_{3,8} = 1.0000000008$ $CTP_{3,9} = 1.0350000008$
Step 3: (a) Generate N random numbers r_1, r_2, r_3, and r_4 in the interval $[0,1]$ without repeating the random number.
$r_1 = 0.3122, r_2 = 0.8701, r_3 = 0.4729, r_4 = 0.6190$
(b). $\boldsymbol{r_a \cong CTP_{i,j}}$ **at path** $\boldsymbol{j}$ **of** $\boldsymbol{X_{i,j}}$ (37)
$r_1 = r_3 = r_4 \equiv CTP_{3,3} \equiv x_3,$ (37a)
$r_2 \equiv CTP_{3,4}$ (37b)
Therefore, the three ants are merge together in the paths then, $Ant1 = Ant2 = Ant3$
Step 4: (a) $\boldsymbol{Ant\ i = f_a = f(x_i) = CTP_{i,j}}$ (38)

$\boldsymbol{Ant}1: r_1 = 0.3122 \cong CTP_{1,3} = 0.8250000008$ then $x_3 = 240$,
Hence, $f(x_1) = f(240) = 2160000$
$\boldsymbol{Ant}2: r_2 = 0.8701 \cong CTP_{1,7} = 0.8888888889$ then $x_7 = 280$,
Hence, $f(x_7) = f(280) = 1680000$
$\boldsymbol{Ant}3: r_3 = 0.4729 \cong CTP_{1,3} = 0.8250000008$ then $x_3 = 240$,
Hence, $f(x_1) = f(240) = 2160000$
$\boldsymbol{Ant}4: r_4 = 0.6190 \cong CTP_{1,4} = 0.8250000008$ then $x_3 = 240$,
Hence, $f(x_1) = f(240) = 2160000$
(b) Determine the $\boldsymbol{f_{best}}$ and $\boldsymbol{f_{worst}}$ (39)
The $\boldsymbol{f_{best}} = f_1 = f_3 = f_4 = f(x_3) = 2160000.$
The $\boldsymbol{f_{worst}} = f_2 = f(x_7) = f(250) = 2100000$
Step 6: (a) $\{\sum_{i=1}^{N} \tau_i\}\rho = \tau_0\, \rho = 0.25 \times 0.5 = 0.125 = \boldsymbol{M_2}$ (40)
(b) $\Delta\tau_{i,k} = \frac{\xi \sum freq\ of\ f_{best}\ (f_{best})}{f_{worst}} = \frac{2 \times 3 \times 2160000}{2100000} = 6.17142857143$
$\Delta\tau_{3,k} = 6.17142857143$
(c) $\tau_3 = M_{i-1} + \sum_{k=1}^{N} \Delta\tau_{2,1} = 0.125 + 6.17142857143 = 6.29642857143$ (41)

Stallion Journal for Multidisciplinary Associated Research Studies
ISSN (Online): 2583-3340
Volume-5 Issue-4 || August 2026 || PP. 1-9
https://doi.org/10.55544/sjmars.5.4.1

Back to step 2

$TP_{4,1} = \frac{M_{i-1}}{(n-freq\ of\ f_{best})(M_{i-1})+\tau_i} = \frac{M_2}{(n-freq\ of\ f_{best})(M_0)+\tau_2}$ (42)

$TP_{4,1} = \frac{0.125}{(9-3)0.25+8} = \frac{0.125}{6\times0.125+6.29642857143} = \frac{0.125}{7.04642857142} = 0.01773948302$

$TP_{4,3}^{jump} = \frac{\tau_3}{(n-freq\ of\ f_{best})(M_1)+\tau_i} = \frac{6.29642857143}{7.04642857142} = 0.89356310188$ (43)

$CTP_{4,1} = TP_{4,1} = 0.01773948302$

$CTP_{4,2} = CTP_{4,1} + TP_{4,1} = 0.03473948302$

$CTP_{4,3} = CTP_{3,2} + TP_{3,3}^{jump} = 0.03473948302 + 0.89356310188 = 0.9283025849$

$CTP_{4,4} = 0.94604206792, CTP_{4,5} = 0.96378155094$

$CTP_{4,6} = 0.98152103396, CTP_{4,7} = 0.99926051698$

$CTP_{4,8} = 1.01700000000, CTP_{4,9} = 1.03473948302$

Step3:
Generate N random numbers r_1, r_2, r_3, and r_4 in the range $[0,1]$ without repeating random number
$r_1 = 0.3122, r_2 = 0.8701, r_3 = 0.4729, r_4 = 0.6190$
(b). $\boldsymbol{r_a \cong CTP_{i,j}}$ **at path $\boldsymbol{j}$ of $\boldsymbol{X_{i,j}}$** (44)
$r_1 = r_2 = r_3 = r_4 \equiv CTP_{4,3} \equiv x_3$, Therefore, the four ants are merge together in the paths i.e $Ant\ 1 = Ant\ 2 = Ant\ 3 = Ant\ 4$
Step 4(a) $\boldsymbol{Ant\ i = f_a = f(x_i) = CTP_{i,j}}$ (45)

Table 1: Summary of the result in Problem 1.

Iteration		**0**	**1**	**2**	**3**
TP		0.11111111111	0.06703397612	0.0350000000	0.01773948302
$\boldsymbol{TP_{jump}}$		Nil	0.56263938081	0.82500000000	0.89356310189
Randomized Numbers	r_1	0.3122	0.3122	0.3122	0.3122
	r_2	0.8701	0.8701	0.8701	0.8701
	r_3	0.4729	0.4729	0.4729	0.4729
	r_4	0.6190	0.6190	0.6190	0.6190
Function Values (× 1,000)	f_1	2160	2160	2160	2160
	f_2	1460	11680	1680	2160
	f_3	2000	2160	2160	2160
	f_4	1860	2160	2160	2160
$\boldsymbol{f_{best}}$		**2160000**	**2160000**	**2160000**	**2160000**
$\boldsymbol{f_{worst}}$		1460000	1680000	168000	268000

$r_1 = 0.3122$, $r_2 = 0.8701$, $r_3 = 0.4729$. and , $r_4 \equiv 0.6190$
$CTP_{4,3} = x_3 = f(x_3) = f(240) = 2160000$ (46)
Unlike the HACO algorithm where the randomized numbers were changed in every iteration, in MACO algorithm, the randomized numbers were kept constant.

Problem 2:
Maximize $f(x) = -75000 + 100x + 0.025x^2 - 0.000004x^3$ at range $4900 \le x \le 6500$.
Number of Ants $= 8, n = 9, \rho = 0.5, \tau_0 = 1, \zeta = 2$.

Table 2: Summary of the result in Problem 2.

Iteration		**0**	**1**	**2**
TP		0.111111111	0.06131259434	0.02615861862
$\boldsymbol{TP_{jump}}$		Nil	0.57081183965	0.8692069089
Randomized Numbers	r_1	0.6710	0.7011	0.9333
	r_2	0.6710	0.7011	0.9333
	r_3	0.6710	0.7011	0.9333

Stallion Journal for Multidisciplinary Associated Research Studies
ISSN (Online): 2583-3340
Volume-5 Issue-4 || August 2026 || PP. 1-9
https://doi.org/10.55544/sjmars.5.4.1

	r_4	0.6710	0.7011	0.9333
	r_5	0.6710	0.7011	
	r_6	0.6710	0.7011	0.9333
	r_7	0.6710	0.7011	0.9333
	r_8	0.6710	0.7011	0.9333
Function Values (× 1,000)	f_1	551326	565750	565750
	f_2	554646	565750	565750
	f_3	561742	565750	565750
	f_4	544654	565750	565750
	f_5	544654	561742	565750
	f_6	565750	561742	565750
	f_7	565750	561742	565750
	f_8	565750	561742	565750
$\boldsymbol{f_{best}}$		565750	565750	565750
$\boldsymbol{f_{worst}}$		544654	561742	

Table 3: The Earliest Ant Colony Algorithms

Algorithms	Iteration	$\boldsymbol{f_{best}}$	$\boldsymbol{f_{worst}}$
ACO	15	585750	544654
AS	10	5.856×10^6	5.45×10^6
MMA	6	5.86×10^6	5.123×10^6
MACO	**2**	**565750**	**544654**

From Tables 1 and 2 it shows that the MACO Algorithm has a better convergence in terms of number of iteration as shown in Table 3.

The optimization problems have been an important problem in the field of distribution and logistics. Since the delivery routes consist of any combination of the customers, this problem belongs to the class of NP-hard problems. In this research, a modified ant colony optimization is proposed for optimization. The Intermittencies Vehicle Routing Problem (IVRP) is a complex and dynamic optimization problem that requires efficient and adaptive solutions.

The MACO algorithm has demonstrated exceptional performance in solving IVRP instances, effectively managing intermittencies and uncertainties.

It takes the advantages of simulated annealing and ant colony optimization. MACO Algorithm, a swarm intelligence based optimization technique, is widely used in network routing. ACO based design optimization is simple, robust and reliable for design optimization problem. The proposed algorithm proved that it has better performance when compared to MACO. The proposed algorithm can be applied to solve IVRP for both small-sized and large-sized instances. MACO's ability to adapt and learn from experience makes it an ideal algorithm for solving IVRP.

IV. CONCLUSION

The development and application of a Modified Ant Colony Optimization (MACO) algorithm with constant variables represents a significant advancement in swarm intelligence research. By addressing the limitations of classical ACO and enhancing algorithm stability through controlled parameterization, the proposed approach offers improved efficiency, reliability, and applicability to complex optimization problems. This study contributes to the growing body of knowledge on modified metaheuristics and provides a foundation for further research in optimization techniques tailored to dynamic and large-scale systems.

REFERENCES

[1] Dorigo, M., & Stützle, T. (2004), Ant Colony Optimization. MIT Press.
[2] Blum, C. (2005), Ant colony optimization: Introduction and recent trends. Physics of Life Reviews, 2(4), 353–373.
[3] Adebayo, K. J., Aderibigbe, F.M., and Dele-Rotimi, A. O., (2019), *"On the Development of a Hybridized Ant Colony Optimization (HACO) Algorithm"*, American Journal of Computational Mathematics, 9, 358-372.

[4] Blum, C., & Roli, A. (2003), Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys, 35(3), 268–308.
[5] Stützle, T., & Hoos, H. H. (2000), MAX–MIN Ant System. Future Generation Computer Systems, 16(8), 889–914.
[6] Aderibigbe, F. M., Adebayo, K. J., and Dele-Rotimi, A., (2015): On Quasi-Newton Method for Solving Unconstrained Optimization Problems, *American Journal of Applied Mathematics,* 3(2), 47-50, http://www.sciencepublishinggroup.com/j/ajam, doi: 10.11648/j.ajam.20150302.13, ISSN: 2330-0043 (Print); ISSN: 2330-006X.
[7] Talbi, E. G. (2009), Metaheuristics: From Design to Implementation. Wiley.
[8] Gendreau, M., & Potvin, J. Y. (2010). Handbook of Metaheuristics. Springer.
[9] Talbi, E. G. (2002), A taxonomy of hybrid metaheuristics. Journal of Heuristics, 8(5), 541–564.
[10] Dorigo, M., Birattari, M., & Stützle, T. (2006), Ant colony optimization: Artificial ants as a computational intelligence technique. IEEE Computational Intelligence Magazine, 1(4), 28–39.
[11] Adebayo, K. J., Aderibigbe, F.M., and Dele-Rotimi, A. O., (2022): On the Developing a Hybridized Ant Colony Optimization (HACO) Algorithm: Some Considerations, *Novel Research Aspects in Mathematical and Computer Science,* Vol. 8, 13-31, DOI: 10.9734/bpi/nramcs/v8/7416F.
[12] Gendreau, M., Laporte, G., & Potvin, J. Y. (2011). Metaheuristics for the vehicle routing problem. European Journal of Operational Research, 211(1), 1–11.
[13] Olaosebikan, T. E., Adebayo, K. J., Dele-Rotimi, A. O., Ayodele, O. J., Popoola, O. E., Akinwumi, T. E., and Egbon, F. O., (2024): On the Solution of Some Optimization Problems Using Free Derivative Based Algorithm, *Journal of Data Acquisition and Processing*, Vol. 39 (1) 2024, 840-845, https://sjcjycl.cn/ doi: 10.5281/zenodo.754738.
[14] Montemanni, R., Gambardella, L. M., Rizzoli, A. E., & Donati, A. V. (2005), Ant colony system for a dynamic vehicle routing problem. Journal of Combinatorial Optimization, 10(4), 327–343.
[15] Olaosebikan, T. E., Olorunsola, S. A., Adebayo, K. J. and (2014), *"On Application of Matrix-Vector Operations for Solving Constrained Optimization Problems",* IOSR Journal of Mathematics (IOSR-JM), 10(3), Ver. IV, 1-5.
[16] Olaosebikan, T. E., Olorunsola, S. A., and Adebayo, K. J., (2014), "*On Application of Lagrange Multiplier for Solving Optimization Problems in both Equality and Inequality Constraints",* IOSR Journal of Mathematics (IOSR-JM), 10(3), Ver. IV, 06-09.