

On the Development of a Hybridized Ant Colony Optimization (HACO) Algorithm

Kayode J. Adebayo^{1*}, Felix M. Aderibigbe¹, Adejoke O. Dele-Rotimi²

¹Department of Mathematics, Ekiti State University, Ado Ekiti, Nigeria

²College of Education, Ikere Ekiti, Nigeria

Email: *kayode.adebayo@eksu.edu.ng, felixaderibigbe@eksu.edu.ng, jokerotimi13@gmail.com

How to cite this paper: 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.
<https://doi.org/10.4236/ajcm.2019.94026>

Received: October 12, 2019

Accepted: December 23, 2019

Published: December 26, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper proposes a Hybridized Ant Colony Optimization (HACO) algorithm. It integrates the advantages of Ant System (AS) and Ant Colony System (ACS) of solving optimization problems. The main focus and core of the HACO algorithm are based on annexing the strengths of the AS, ACO and the Max-Min Ant System (MMAS) previously proposed by various researchers at one time or the order. In this paper, the HACO algorithm for solving optimization problems employs new Transition Probability relations with a Jump transition probability relation which indicates the point or path at which the desired optimum value has been met. Also, it brings to play a new pheromone updating rule and introduces the pheromone evaporation residue that calculates the amount of pheromone left after updating which serves as a guide to the successive ant traversing the path and diverse local search approaches. Regarding the computational efficiency of the HACO algorithm, we observe that the HACO algorithm can find very good solutions in a short time, as the algorithm has been tested on a number of combinatorial optimization problems and results shown to compare favourably with analytical results. This strength can be combined with other metaheuristic approaches in the future work to solve complex combinatorial optimization problems.

Keywords

Ant Colony System, Metaheuristics, Pheromone, Jump Transition Probability, Pheromone Evaporation Residue, Hybridized Ant Colony Optimization

1. Introduction

Ant System algorithms have been founded on the observation of real ant colonies by various researchers to which Marco Dorigo a key factor as it was found

in his Ph.D. Thesis of 1992, where the concept was first brought to light. By living in colonies, ants' social behavior is directed more to the survival of the colony than that of a single individual member of the colony. An interesting and significantly important behavior of ant colonies is their foraging behavior, and in particular, their ability to find the shortest route between their nest and a food source, realizing that they are almost blind. Without any leader that guides the ants to optimal trajectories, the ants manage to find these optimal trajectories with time, by interacting with their local environment.

The ants initially search for food in a random fashion such that, when they have found some, they return home depositing chemicals, called pheromones. According to [1], these pheromones attract other ants to follow the same path, and they in turn also deposit pheromones on their way back. Over a period of time, this behavior leads to the emergence of paths, which can be shown to be near-optimal which is essentially done at random. By smelling the pheromone, there is a higher probability that the trail with a higher pheromone concentration will be chosen. The pheromone trail allows ants to find their way back to the food source and vice versa.

It follows that when a number of paths are available from the nest to a food source, a colony of ants may be able to exploit the pheromone trail deposited by the individual members of the colony to discover the shortest path from the nest to the food source and back [2]. As more ants choose a path to follow, the pheromone on the path builds up, making it more attractive to other ants seeking food and hence more likely to be followed by other ants.

In general, Ant Colony Optimization (ACO) algorithms employ a finite size of artificial agents with defined characteristics which collectively search for good quality solutions to the problem under consideration. Starting from an initial state selected according to some case-dependent criteria, each ant builds a solution, which is similar to a chromosome in a genetic algorithm. While building its own solution, each ant collects information on its own performance and uses this information to modify the representation of the problem, as seen by the other ants [3]. The ant's internal states store information about the ant's past behavior, which can be employed to compute the goodness/value of the generated solution. Artificial ants are permitted to release pheromone while developing a solution or after a solution has fully been developed, or both.

The amount of pheromone deposited is made proportional to the goodness of the solution an artificial ant has developed (or is developing). Rapid drift of all the ants towards the same part of the search space is avoided by employing the stochastic component of the choice decision policy and the pheromone evaporation mechanism. To simulate pheromone evaporation, the pheromone persistence coefficient, ρ , is defined which enables greater exploration of the search space and minimizes the chance of premature convergence to suboptimal solutions.

A probabilistic decision policy is also used by the ants to direct their search towards the most interesting regions of the search space. The level of stochastic-

ity in the policy and the strength of the updates in the pheromone trail determine the balance between the exploration of new points in the state space and the exploitation of accumulated knowledge [3].

This paper is structured as follows: Section 2 describes and briefly reviews variants Ant systems that have been developed over years by various researchers. While Section 3 is dedicated to the introduction and description in detail of the Hybridized Ant Colony Optimization (HACO) algorithm. Section 4 presents a step by step computational procedure of the HACO algorithm. Section 5 concludes the paper.

2. Ant System Variants

The concept of metaheuristic is a general purposes algorithmic framework that can be applied to different optimization problems with relatively few modifications to make them adaptable to a specific problem. In the view of [3], it was observed that, when adapting an Ant Colony Optimization approach to a particular problem, the following choices need to be adapted:

- A) How do the ants construct a solution?
- B) How do the ants choose the next step to take?
- C) How is the solution post-optimized?
- D) How is the pheromone deposit updated?

Any variant of ACO algorithm that will be regathered as amenable will find answers to all the afore mentioned questions. There are numerous examples of Combinatorial Optimization problems and the variants being handled using Ant Colony Optimization algorithms. This is a type of algorithm that seeks to model the emergent behaviour observed in ant colonies and utilize this behaviour to solve problems. Before we roll out the HACO algorithm there is need to have an outline to various ACO developed by various researchers. This leads us to variants ACO algorithms that have been developed over years by various researchers and improvements that ACO algorithms have witnessed.

In the last three decades, there have been variants Ant Colony Optimizations due to modification and improvements in one respect or the other. It must be noted that, the several variants Ant Colony Optimization differing most importantly on when and how pheromones are updated. However, some of the variants are highlighted and discussed in what follows.

2.1. Ant System (AS)

Ant Colony Optimization (ACO), called Ant System in [4] was inspired by the studies and the behaviour of ants [5]. The ant-colony metaheuristic framework was introduced by [1], which enabled ACO to be applied to a range of combinatorial optimization problems. The authors in [1] also reported the successful application of ACO algorithms to a number of bench-mark combinatorial optimization problems. ACO in [3], is “a metaheuristic that is inspired by the pheromone trail laying and following behaviour of some ant species”.

Ant System is the first ACO algorithm proposed in literature. The AS is based on the behavior of real ants searching for food. Real ants communicate with each other using an aromatic essence, called pheromone, that they lay down on the path they traverse. By [6], the pheromone accumulates when more and more ants pass through the same path. Nevertheless, the pheromone will evaporate if no ants continue to pass. The selection of the pheromone trail reflects the length of the paths as well as the quality of the food source found. [7] reported the use of AS to solve Traveling Salesman Problem (TSP), Quadratic Assignment Problems (QAP) and Job-Shop Scheduling.

Its main characteristics are as follows: at each iteration, the pheromone values are updated by all the m ants that have built a solution in the iteration itself. At each step, an ant at node i chooses to go to a node j through the edge (i, j) and this arc is added to the solution. The repetition of this step stops when the ant has completed its tour. The pheromone, τ_{ij} , associated with the edge joining cities, i and j , is updated using the relation:

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

where ρ is the evaporation rate, m is the number of ants, and $\Delta \tau_{ij}^k$ is the quantity of pheromone laid on edge (i, j) by ant k :

$$\Delta \tau_{ij}^k = \begin{cases} 1/L_k & \text{if ant } k \text{ used } (i, j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where L_k is the length of the tour constructed by ant k .

2.2. Ant Colony System (ACS)

The ant colony system (ACS) was developed as in [8] and [9] to improve the performance of AS. A different state transition rule was used by [5] to which a local pheromone updating rule was also added. [2] Proposed an ACS algorithm based on AS. The first difference between the ACS algorithm and the AS concerns the update of the pheromone trail. The update done at the end of an iteration of the algorithm is called offline.

Once all the ants have built a solution, a pheromone trail is added to the arcs used by the ant that has found the best tour from the beginning of the trial. Thus, instead of allowing all the m ants to update the pheromones as in AS, only the ant that has found the best solution in ACS deposits pheromone on the arcs of the best tour. The local pheromone update is performed by all the ants after each construction step. Each ant applies update only to the last edge traversed as opined by [10] thus:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\tau_0 \quad (3)$$

where $\rho \in (0, 1]$ is the pheromone decay coefficient, and τ_0 is the initial value of the pheromone.

The offline pheromone update, similarly to Max-Min Ant System (MMAS), is applied at the end of each iteration by the ant with the shortest tour, which can be either the iteration-best or the best-so-far. The update formula is:

$$\tau_{ij} \leftarrow \begin{cases} (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}, & \text{if } (i, j) \in t \\ \tau_{ij}, & \text{otherwise} \end{cases} \quad (4)$$

In MMAS, not all pheromone levels are evaporated, as with the AS, but only those that also receive a pheromone deposit.

2.3. MAX-MIN Ant System (MMAS)

In MMAS, only one single ant is used to update the pheromone trails after every iteration. The algorithm achieves a strong exploitation of the search history by allowing only the best solutions to add pheromone during the pheromone trail update. Its characterizing elements are that only the best ant updates the pheromone trails and that the value of the pheromone is bounded by $|\tau_{\min} - \tau_{\max}|$. Also, the use of a rather simple mechanism for limiting the strengths of the pheromone trails effectively avoids premature convergence of the search.

The pheromone update is implemented as follows:

$$\tau_{ij} = \left[(1-\rho) \cdot \tau_{ij} + \tau_{ij}^{best} \right]_{\tau_{\min}}^{\tau_{\max}} \quad (5)$$

where τ_{\max} and τ_{\min} are respectively the upper and lower bounds imposed on the pheromone; the operator $[x]_b^a$ is defined as:

$$[x]_b^a = \begin{cases} a & \text{if } x > a \\ b & \text{if } x < b \\ x & \text{otherwise} \end{cases} \quad (6)$$

and τ_{ij}^{best} is defined by:

$$\tau_{ij}^{best} = \begin{cases} 1/L_{best} & \text{if } (i, j) \text{ belongs to the best tour} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where L_{best} is the length of the tour of the best ant. This may be either the best tour found in the current iteration *i.e.* iteration-best, L_{ib} or the best solution found since the start of the algorithm-*best-so-far*, L_{bs} or a combination of both [10] and [11]. We observe that quite a number of parameters need to be modified and improved upon.

The Hybridized Ant Colony Optimization hinges on the strengths of the aforementioned variants and improves on the drawbacks leading to the hybridized algorithm stressing on the Transition Probability, Jump Transition Probability, generating Randomized Numbers that guides in determining the path traversed by the ants, pheromone updating rule, pheromone evaporation residue and diverse local search approaches are proposed hence, evaluating the function values to which the best and worst function values is identified will be discussed in the next section.

3. Hybridized Ant Colony Optimization (HACO) Algorithm

The HACO algorithm is described as follows:

Step 1: Initialize, N, n, ρ, τ_0 . Set $\xi = 0.00001$ and $\zeta = 2$.

where $A_a, a = 1, 2, \dots, N$ represent the number of Ants in the colony. $\{x_j\}$

represents a set of permissible discrete values of the Design Variables, x_j , and $j = 1, 2, \dots, n$. The interval, $\alpha \leq x_j \leq \beta$ or $[\alpha, \beta]$, is divided into n equal parts such that, each part is assigned to each of the design variables, x_j . The pheromone evaporation rate is assumed to be $\rho = 0.5$. It is assumed that all the ants will deposit equal amount of pheromone at the initial stage hence, $\tau_0 = 1$.

Step 2: a) Compute the Transition Probability, $TP_{i,j}$, using:

$$TP_{i,j} = \frac{\tau_i}{(N - \sum \text{freq of } f_{best})(\tau_i) + \tau_{i+1}} \quad (8)$$

where f_{best} is the highest function value, $\sum \text{freq of } f_{best}$ is the sum of the frequency of the f_{best} where more than one exist and the Jump Transition Probability, $P_{i,j}^{jump}$ is computed using:

$$TP_{i,j}^{jump} = \frac{\tau_{i+1}}{(N - \sum \text{freq of } f_{best})(\tau_i) + \tau_{i+1}} \quad (9)$$

where the subscripts, i and j of $TP_{i,j}$ and $TP_{i,j}^{jump}$ are the iterations and paths counters respectively. The cumulative transition probabilities' values are computed and it is being associated with the paths traversed by the ant based on the value of the transition probability obtained using the expressions (8) and (9).

N.B: At $i = 0$, the Jump Transition Probability, $TP_{i,j}^{jump}$, is not computed since each path has equiprobable chances of being traversed. Also, the $\sum \text{freq of } f_{best} = 0$ and $\tau_{i+1} = 0$. The Jump Transition Probability, $TP_{i,j}^{jump}$, is aimed at speeding up the transition probability of the path traversed after a path have been identified as being best owing to the function value.

b) Compute the Cumulative Transition Probability, $CTP_{i,j}$.

At iteration $i = 0$, the Cumulative Transition Probability, $CTP_{i,j}$ along the path, j , is given by:

$$\left. \begin{aligned} CTP_{i,1} &= TP_{i,1} \\ CTP_{i,2} &= CTP_{i,1} + TP_{i,1} \\ CTP_{i,3} &= CTP_{i,2} + TP_{i,1} \\ &\vdots \\ CTP_{i,n} &= CTP_{i,n-1} + TP_{i,1} \end{aligned} \right\} \quad (10)$$

At iteration $i > 0$, the Cumulative Transition Probability, $CTP_{i,j}$ along the path, j , is given by:

$$\left. \begin{aligned} CTP_{i,1} &= TP_{i,1} \\ CTP_{i,2} &= CTP_{i,1} + TP_{i,1} \\ &\vdots \\ CTP_{i,m} &= CTP_{i,m-1} + TP_{i,1} \\ CTP_{i,m+1} &= CTP_{i,m} + TP_{i,j}^{jump} \\ CTP_{i,m+2} &= CTP_{i,m+1} + TP_{i,1} \\ CTP_{i,m+3} &= CTP_{i,m+2} + TP_{i,1} \\ &\vdots \\ CTP_{i,n} &= CTP_{i,(n-1)} + TP_{i,1} \end{aligned} \right\} \quad (11)$$

Step 3: a) Generate N random numbers.

The specific path chosen by a particular ant will then be determined using the Roulette Wheel selection process of randomization. From the closed interval $[0,1]$, N random numbers:

$$r_1, r_2, \dots, r_N \quad (12)$$

are generated such that, an ant, A_a , is assigned a random number each. The assigned randomized number is then further used as a guide towards the path traversed by the ant.

b) Correlate each of the randomized numbers, r_a .

Correlate each of the randomized numbers, r_a , with a transition probability cumulative value, $CTP_{i,j}$, such that the randomized number, r_a , falls within the range of value of the $CTP_{i,j}$ along a particular path thus:

$$r_a \approx CTP_{i,j} \quad (13)$$

Step 4: a) Evaluate the objective function value, $f(x_j)$.

The objective function value, $f(x_j)$ is evaluated with the value assigned to the design variable, x_j , that corresponds to the cumulative transition probability's paths, j , of the design variable path. *i.e.*

$$f_a = CTP_{i,j}(j) = f(x_j) \quad (14)$$

b) Determine the best and worst paths.

The best and worst paths among the N paths chosen by different ants are determined using the relation:

$$f_{best} = \text{Max}(f(x_1), f(x_2), \dots, f(x_N)) \quad (15)$$

$$f_{worst} = \text{Min}(f(x_1), f(x_2), \dots, f(x_N)) \quad (16)$$

Step 5: Test for convergence of the process.

The process is assumed to have converged if all the ants take the same path. *i.e.*

$$f(x_1) = f(x_2) = \dots = f(x_N) \quad \text{or} \quad (17a)$$

$$f(x_N) - f(x_{N-1}) = f(x_{N-1}) - f(x_{N-2}) = \dots = f(x_2) - f(x_1) = \xi \leq 0.00001 \quad (17b)$$

If the convergence criterion has been satisfied, then stop. Else, it will be assumed that, all the ants return to the colony and start the search for a shorter path to the food source again.

Step 6: a) Calculate the Pheromone Residue, M :

$$M = \left(\sum_{i=1}^N \tau_i \right) - \rho \quad (18)$$

where ρ , rate of pheromone evaporation, is a user defined parameter.

b) Evaluate the objective function value's ratio

The range ratio of the best objective function value, f_{best} , to the worst objective function value, f_{worst} , in maximizing a function given by $\Delta \tau_{jk}$, is evaluated using the relation:

$$\Delta \tau_{jk} = \frac{\zeta (\sum \text{freq of } f_{best})(f_{best})}{f_{worst}} \quad (19)$$

where the summation extends over all the best ants, A_a (if multiple ants take the same best path). The subscripts, j , is the path preceding node and k is the path succeeding node.

c) Compute the pheromone deposit update:

To compute the pheromone deposits update set $i = i + 1$ and use the following relations:

$$\tau_i = (1 - \rho) \tau_{i-1} \quad (20)$$

$$\tau_{i+1} = \tau_i + \Delta \tau_{jk} \quad (21)$$

where τ_{i+1} is the pheromone on the node link, (j, k) , after updating and τ_i is the pheromone on the link (j, k) . This is before the update is computed, then, go to Step 2.

4. Test Problems

The HACO algorithm is now applied to solve the following test problems:

Problem 1: Find the Maximum of

$$f(x) = -200x^2 + 92000x - 8400000 \quad (22)$$

within the range 220 - 300 using HACO algorithm.

Step 1: Let $A_a, a = 1, 2, \dots, N$ represent the number of Ants in the colony.

Let $\{x_j\}$ represent a set of permissible discrete values of the Design Variable, x_j , where $j = 1, 2, \dots, n$. The range, $\alpha \leq x_j \leq \beta$ or $[\alpha, \beta]$, is divided into n equal parts such that, a part each is assigned to each of the design variables, x_j . Let $\rho = 0.5$ represent the pheromone evaporation rate. It is assumed that all the ants will deposit equal amount of pheromone at the initial stage hence, $\tau_1 = 1$.

Number of Ants = 4, $n = 9$, are assumed within the range for x_i as:

$$\begin{aligned} x_1 &= 220, & x_2 &= 230, & x_3 &= 240, \\ x_4 &= 250, & x_5 &= 260, & x_6 &= 270, \\ x_7 &= 280, & x_8 &= 290, & x_9 &= 300 \end{aligned}$$

Step 2: Compute the Transition Probability, $TP_{i,j}$, of selecting the arc or ray (or the discrete values) x_j as:

$$TP_{i,j} = \frac{\tau_i}{(N - \sum \text{freq of } f_{best})(\tau_i) + \tau_{i+1}}, \quad j = 1, 2, \dots, n \quad (23)$$

$$TP_{1,1} = \frac{1}{(9-0) \times 1 + 0} = \frac{1}{9} = 0.1111111111 \quad (24)$$

The cumulative of the transition probability values are computed associated with different paths based on the transition probabilities values obtained from the expressions (8) and (9). The Cumulative Transition Probability, $CTP_{i,j}$ at iteration i and path j is given by:

$$\left. \begin{aligned} CTP_{i,1} &= TP_{i,1} \\ CTP_{i,2} &= CTP_{i,1} + TP_{i,1} \\ CTP_{i,3} &= CTP_{i,2} + TP_{i,1} \\ &\vdots \\ CTP_{i,n} &= CTP_{i,n-1} + TP_{i,1} \end{aligned} \right\}$$

Hence,

$$\begin{aligned} CTP_{1,2} &= 0.1111111111 + 0.1111111111 = 0.2222222222, \\ CTP_{1,3} &= 0.1111111111 + 0.2222222222 = 0.3333333333, \\ CTP_{1,4} &= 0.1111111111 + 0.3333333333 = 0.4444444444, \\ CTP_{1,5} &= 0.1111111111 + 0.4444444444 = 0.5555555555, \\ CTP_{1,6} &= 0.1111111111 + 0.5555555555 = 0.6666666666, \\ CTP_{1,7} &= 0.1111111111 + 0.6666666666 = 0.7777777777, \\ CTP_{1,8} &= 0.1111111111 + 0.7777777777 = 0.8888888888, \\ CTP_{1,9} &= 0.1111111111 + 0.8888888888 = 1.0000000000. \end{aligned}$$

N. B: Since none of the paths have been identified as f_{best} , the jump transition probability needs not be computed at this stage.

Step 3: a) Generate N random numbers, r_1, r_2, \dots, r_N , within the $[0,1]$. One random number for each ant thus:

$$\begin{aligned} r_1 &= 0.3122, \quad r_2 = 0.8701, \\ r_3 &= 0.4729, \quad r_4 = 0.6190 \end{aligned} \tag{25}$$

b) Correlate each of the randomized numbers, r_a , with a transition probability cumulative value, $CTP_{i,j}$, which is of the most approximate value or the cumulative transition probability value's range to which the randomized numbers value, r_a , fall:

$$r_a \cong CTP_{i,j}. \tag{26}$$

Ant 1: $r_1 = 0.3122 \approx CTP_{1,3} = 0.3333333333$

Ant 2: $r_2 = 0.8701 \approx CTP_{1,8} = 0.8888888888$

Ant 3: $r_3 = 0.4729 \approx CTP_{1,5} = 0.5555555555$

Ant 4: $r_4 = 0.6190 \approx CTP_{1,6} = 0.6666666666$

Step 4: a) Evaluate the objective function value, $f(x_j)$.

The objective function values, $f(x_j)$ are evaluated using values assigned to the design variable values, x_j , corresponding to the cumulative of the transition probability's paths, j .

$$f_a = CTP_{i,j} \text{ at } j = f(x_j) \tag{27}$$

Ant 1: $f_1 = CTP_{1,3} = 0.3333333333$.

Then, $x_3 = 240$. Hence, $f(x_1) = f(240) = 2160000$.

Ant 2: $f_2 = CTP_{1,8} = 0.8888888888$.

Then, $x_8 = 290$. Hence, $f(x_2) = f(290) = 1460000$.

Ant 3: $f_3 = CTP_{1,5} = 0.555555555556$.

Then, $x_5 = 260$. Hence, $f(x_3) = f(260) = 2000000$.

Ant 4: $f_4 = CTP_{1,6} = 0.666666666667$.

Then, $x_6 = 270$. Hence, $f(x_4) = f(270) = 1860000$.

b) Determine the best and worst paths.

The best and worst paths among the N paths chosen by different ants are determined using the relation:

$$f_{best} = \text{Max}(f(x_1), f(x_2), \dots, f(x_N)) \quad (28)$$

$$f_{worst} = \text{Min}(f(x_1), f(x_2), \dots, f(x_N)) \quad (29)$$

Hence, $f_{best} = f(x_3) = 216000$ at $x_{best} = x_3 = 240$ and

$f_{worst} = f(x_8) = 1460000$ at $x_{worst} = x_8 = 290$.

Step 5: Test for convergence of the process. The process is assumed to have converged if all the N ants take the same path. *i.e.*

$$f(x_1) = f(x_2) = \dots = f(x_N) \quad \text{or} \quad (30)$$

$$f(x_N) - f(x_{N-1}) = f(x_{N-1}) - f(x_{N-2}) = \dots = f(x_2) - f(x_1) = \xi \leq 0.00001 \quad (31)$$

However, if convergence has not been achieved, it will be assumed that all the ants will have to return home and start again in search of food. Since neither of the convergence conditions is met, the process has to continue.

Step 6: a) Calculate the Pheromone Evaporation constant, M .

Since the convergence criterion has not been met, there is need to calculate the pheromone evaporation constant, M , using the relation:

$$M = \left(\sum_{i=1}^N \tau_i \right) - \rho. \quad (32)$$

$$M = 4 \times 1 - 0.5 = 3.5$$

b) Evaluate the objective function value's ratio:

$$\Delta \tau_{jk} = \frac{\zeta (\sum \text{freq of } f_{best})(f_{best})}{f_{worst}} \quad (33)$$

$$\Delta \tau_{2,4} = \frac{1 \times 2 \times 2160000}{1460000} = 2.95890410959$$

where $\zeta = 2, \rho = 0.5$.

c) Compute the Pheromone deposit update: Then, set the new iteration number as $i = i + 1$, and update the pheromone deposits on different arcs (or discrete values of design variables) where τ_{i-1} denotes the previous pheromone amount of the iteration after evaporation, which is taken as τ_i can be computed by:

At $i = i + 1 = 2$,

$$\tau_i = (1 - \rho) \tau_{i-1}$$

$$\tau_2 = (1 - 0.5) \tau_1 = 0.5 \times 1.0 = 0.5$$

$$\tau_{i+1} = \tau_i + \Delta \tau_{jk}, \tag{34}$$

where $j = 1, 2, 4, 5, 6, 7, 8, 9$. It must be noted that the path, $j = 3$ is omitted. This is because, that was the path at which f_{best} has been identified.

$$\tau_3 = \tau_2 + \Delta \tau_{2,4} = 1.0 + 2.95890410959 = 3.95890410959$$

With this, we go back to step 2.

Step 2: Compute the Transition Probability, $(TP_{i,j})$ and Transition Probability Jump, $(TP_{i,j}^{jump})$ of selecting the arc:

$$TP_{i,j} = \frac{\tau_i}{(N - \sum \text{freq of } f_{best})(\tau_i) + \tau_{i+1}} \text{ and}$$

$$TP_{i,j}^{jump} = \frac{\tau_{i+1}}{(N - \sum \text{freq of } f_{best})(\tau_i) + \tau_{i+1}}$$

where $\text{freq of } f_{best}$ is the frequency of the f_{best} , $\tau_2 = 0.5$, $\tau_3 = 3.95890410959$ and $j = 1, 2, 4, 5, 6, 7, 8, 9$

This gives:

$$TP_{2,1} = \frac{\tau_2}{(N - \sum \text{freq of } f_{best})(\tau_2) + \tau_3}$$

$$= \frac{0.5}{(9 - 1) \times 0.5 + 3.95890410959}$$

$$= 0.06282271945$$

$$TP_{2,3}^{jump} = \frac{\tau_3}{(N - \sum \text{freq of } f_{best})(\tau_2) + \tau_3}$$

$$= \frac{3.95890410959}{7.95890410959} = 0.49741824441$$

The Transition Probabilities would have been equiprobable but for the fact that, the transition probability has to jump at $P_{2,3}$. Hence,

$$CTP_{2,1} = 0.06282271945,$$

$$CTP_{2,2} = 0.06282271945 + 0.06282271945 = 0.1256454389,$$

Note that, there is a jump at this point. Hence,

$$CTP_{2,2}^{jump} = 0.49741824441 + 0.12564543890 = 0.62306368330,$$

$$CTP_{2,4} = 0.06282271945 + 0.62306368330 = 0.68588640276,$$

$$CTP_{2,5} = 0.06282271945 + 0.68588640276 = 0.74870912221,$$

$$CTP_{2,6} = 0.06282271945 + 0.74870912221 = 0.81153184166,$$

$$CTP_{2,7} = 0.06282271945 + 0.81153184166 = 0.87435456111,$$

$$CTP_{2,8} = 0.06282271945 + 0.87435456111 = 0.93717728056,$$

$$CTP_{2,9} = 0.06282271945 + 0.93717728056 = 1.00000000001$$

Step 3: Generate N random numbers r_1, r_2, \dots, r_N in the $[0,1]$, one for each ant

$$r_1 = 0.3688, r_2 = 0.8577, r_3 = 0.0776, r_4 = 0.5791$$

a) Evaluate the objective function values corresponding to the complete paths

$$\text{Ant 1: } f_1 = f(x_1) = f(240) = 2160000$$

$$\text{Ant 2: } f_2 = f(x_2) = f(280) = 1680000$$

$$\text{Ant 3: } f_3 = f(x_3) = f(230) = 2180000$$

$$\text{Ant 4: } f_4 = f(x_4) = f(240) = 2160000$$

Step 4: Determine the best and worst paths among the N paths chosen by different ants

$$x_{best} = x_3 = 230; f_{best} = f_3 = 2180000$$

$$x_{worst} = x_2 = 280; f_{worst} = f_2 = 1680000$$

Step 5: At $i = 3$, $\tau_{i+1} = \tau_i + \Delta\tau_{jk}$,

$$\Delta\tau_{jk} = \frac{\zeta (\sum \text{freq of } f_{best})(f_{best})}{f_{worst}} = \frac{2 \times 2180000}{1680000} = 2.59523809524,$$

where $\zeta = 2, \rho = 0.5$ and τ_i can be computed as:

$$\tau_i = (1 - \rho)\tau_{i-1} = (1 - 0.5)\tau_{i-1} = 0.5 \times 0.5 = 0.25; j = 1, 3, 4, 5, 6, 7, 8, 9$$

with the path 2 missed out. Hence,

$$\tau_4 = \tau_3 + \Delta\tau_{1,3} = 3.95890410959 + 2.59523809524 = 6.55414220483$$

for $j = 2$ and $\tau_3 = 0.25$

With this, we set $i = 4$ and go back to step 2.

Step 2: Compute the Transition Probability (TP_{ij}) of selecting the arc

$$TP_{i,j} = \frac{\tau_i}{(N - \sum \text{freq of } f_{best})(\tau_i) + \tau_{i+1}} \text{ and}$$

$$TP_{i,j}^{jump} = \frac{\tau_{i+1}}{(N - \sum \text{freq of } f_{best})(\tau_i) + \tau_{i+1}},$$

where $\tau_3 = 0.25$; $j = 1, 3, 4, 5, 6, 7, 8, 9$ and $\tau_4 = 6.55414220483$

$$\text{This gives } TP_{3,1} = \frac{0.25}{8 \times 0.25 + 6.55414220483} = \frac{0.25}{8.55414220483}$$

$$TP_{3,1} = 0.0292256072$$

$$TP_{3,2}^{jump} = \frac{6.55414220483}{8.55414220483} = 0.76619514241$$

$$CTP_{3,3} = 0.82464635681, CTP_{3,4} = 0.85387196401,$$

$$CTP_{3,5} = 0.88309757121, CTP_{3,6} = 0.91232317841,$$

$$CTP_{3,7} = 0.94154878561, CTP_{3,8} = 0.97077439281, CTP_{3,9} = 1.00000000001$$

Step 3: a) Generate N random numbers r_1, r_2, \dots, r_N in the $[0, 1]$, one for each ant:

$$r_1 = 0.9251, r_2 = 0.4928, r_3 = 0.0632, r_4 = 0.2064$$

b) Evaluate the objective function values corresponding to the complete paths:

$$\text{Ant 1: } f_1 = f(x_1) = f(280) = 1680000$$

$$\text{Ant 2: } f_2 = f(x_2) = f(230) = 2180000$$

$$\text{Ant 3: } f_3 = f(x_3) = f(230) = 2180000$$

$$\text{Ant 4: } f_4 = f(x_4) = f(230) = 2180000$$

Step 4: Determine the best and worst paths among the N paths chosen by different ants

$$x_{best} = x_{1,2} = 230; f_{best} = f_2 = f_3 = f_4 = 2180000$$

$$x_{worst} = x_{1,7} = 280; f_{worst} = f_1 = 1680000$$

Step 5: $i = 4$,

$$\Delta\tau_{1,5} = \frac{\zeta(\sum \text{freq of } f_{best})(f_{best})}{f_{worst}} = \frac{3 \times 2 \times 2180000}{1680000} = 7.78571428571$$

where $\zeta = 2, \rho = 0.5$, τ_i and τ_{i+1} can be computed as

$$\tau_i = (1 - \rho)\tau_{i-1} = (1 - 0.5)\tau_3 = 0.5 \times 0.25 = 0.125; j = 1, 5, 6, 7, 8, 9$$

$$\tau_{i+1} = \tau_i + \Delta\tau_{jk}$$

$$\tau_5 = \tau_4 + \Delta\tau_{1,5} = 6.55414220483 + 7.7857142857 = 4.33985649054$$

for $j = 2$ and $\tau_4 = 0.125$

With this we back go to step 2.

Step 2: Compute the Cumulative transition probability (CTP_{ij}) of selecting the arc:

$$TP_{i,j} = \frac{\tau_i}{(N - \sum \text{freq of } f_{best})(\tau_i) + \tau_{i+1}} \text{ and}$$

$$TP_{i,j}^{jump} = \frac{\tau_{i+1}}{(N - \sum \text{freq of } f_{best})(\tau_i) + \tau_{i+1}},$$

where $\tau_{ij} = 0.125; j = 1, 3, 4, 5, 6, 7, 8, 9$ and $\tau_{4,2} = 14.33985649054$

This gives:

$$TP_{4,1} = \frac{0.125}{6 \times 0.125 + 14.33985649054} = \frac{0.125}{15.08985649054} = 0.00828371032$$

$$TP_{4,2}^{jump} = \frac{14.33985649054}{15.08985649054} = 0.95029773806$$

$$CTP_{4,1} = 0.00828371032,$$

$$CTP_{4,2}^{jump} = 0.95858144838, CTP_{4,3} = 0.9668651587,$$

$$CTP_{4,4} = 0.97514886902, CTP_{4,5} = 0.98343257934,$$

$$CTP_{4,6} = 0.99171628966, CTP_{4,7} = 0.99999999998,$$

$$CTP_{4,8} = 1.0082837103, CTP_{4,9} = 1.01656742062$$

Step 3: a) Generate N random numbers r_1, r_2, \dots, r_N in the $[0, 1]$, one for each ant:

$$r_1 = 0.5684, r_2 = 0.8385,$$

$$r_3 = 0.4191, r_4 = 0.250$$

b) Evaluate the objective function values corresponding to the complete paths:

$$\text{Ant 1: } f_1 = f(x_1) = f(230) = 2180000$$

$$\text{Ant 2: } f_2 = f(x_2) = f(230) = 2180000$$

$$\text{Ant 3: } f_3 = f(x_3) = f(230) = 2180000$$

$$\text{Ant 4: } f_4 = f(x_4) = f(230) = 2180000$$

It can be seen that all the ants now follow the same path, that is, Ant 1, Ant 2, Ant 3 and Ant 4 follow the same path. Hence, this is acclaimed as the best path with:

$$x_{best} = x_2 = 230, f_{best} = f_1 = f_2 = f_3 = f_4 = 2180000$$

$$\tau_{4,2} = 14.33985649054 \quad \text{and} \quad f(x^*) = 2180000.$$

5. Conclusion

In this paper, we tried to annex the strengths of AS, ACS and MMAS, Ant Colony Optimization algorithms, to obtain solution of optimization problems. To this means, due to high computational complexity in solving these problems using the earliest ACO algorithms, we modify the relation for determining the Transition Probability, formulate the Jump Transition Probability and Pheromone Evaporation Residue relations and introduce the process of correlating the randomized numbers, r_a , with the cumulative transition probability cumulative value, $CTP_{i,j}$, into the algorithm to obtain the best solution to build a Hybridized ACO such that the randomized number, r_a , falls within the range of value of the $CTP_{i,j}$ along a particular path. Implementation of the method and numerical results that show the accuracy of the method, speedy convergence and low occupation of the computer memory are also discussed.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Colorni, A., Dorigo, M. and Maniezzo, V. (1991) Distributed Optimization by Ant Colonies. In: Varela, F. and Bourgine, P., Eds., *Proceedings of ECAL91—European Conference on Artificial Life*, Elsevier Publishing, Paris, France, 134-142.
- [2] Dorigo, M., Birattari, M. and Stützle, T. (2006) Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique. Technical Report, IRIDIA, Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle, Université Libre de Bruxelles.
- [3] Dorigo, M. and Di Caro, G. (1999) The Ant Colony Optimization Meta-Heuristic. In: Corne, D., Dorigo, M. and Glover, F., Eds., *New Ideas in Optimization*, McGraw-Hill, London, UK, 11-32.

- [4] Dorigo, M. (1992) Optimization, Learning and Natural Algorithms, Ph.D. Thesis, DEI, Politecnico di Milano, Italy.
- [5] Dorigo, M. and Gambardella, L.M. (1996) A Study of Some Properties of Ant-Q. In: Voigt, H.-M., Ebeling, W., Rechenberg, I. and Schwefel, H.-S., Eds., *Proceedings of PPSN IV—Fourth International Conference on Parallel Problem Solving From Nature*, Springer-Verlag, Berlin, 656-665. https://doi.org/10.1007/3-540-61723-X_1029
- [6] Chen, C.-H. and Ting, C.-J. (2006) An Improved Ant Colony System Algorithm for the Vehicle Routing Problem. *Journal of the Chinese Institute of Industrial Engineers*, **23**, 115-126. <https://doi.org/10.1080/10170660609509001>
- [7] Dorigo, M., Maniezzo, V. and Colorni, A. (1996) Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, **26**, 29-41. <https://doi.org/10.1109/3477.484436>
- [8] Dorigo, M. and Gambardella, L.M. (1997) Ant Colonies for the Traveling Salesman Problem. *Biosystems*, **43**, 73-81. [https://doi.org/10.1016/S0303-2647\(97\)01708-5](https://doi.org/10.1016/S0303-2647(97)01708-5)
- [9] Dorigo, M. and Gambardella, L.M. (1997) Ant Colony System: A Cooperative Learning Approach for the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, **1**, 53-66. <https://doi.org/10.1109/4235.585892>
- [10] Russell, S.J. and Norvig, P. (2009) Artificial Intelligence: A Modern Approach. 3rd Edition, Prentice Hall, Bergen, NJ.
- [11] Florence, M. (2013) Optimization Approaches for Vehicle Routing Problems with Black Box Feasibility. A Thesis Submitted to Institute of Information and Communication Technologies, Electronics and Applied Mathematics Louvain School of Engineering, Louvain-La-Neuve, Belgium.