ISSN 1391-0256

Journal of the Sri Lanka Association for the Advancement of Science

Volume 8 Issue 1, 2026

JSLAAS

Founded in 1944 and incorporated by the Act of Parliament No 11 of 1966

Journal of the Sri Lanka Association for the Advancement of Science is a biannual publication. Selected research work from annual research sessions (based on scientific merit) as well as other research articles are invited to submit research manuscripts as per the guidelines provided by SLAAS. SLAAS members may also separately submit their papers for publication. The Journal can be accessed on-line to view and download the full text of the articles published respective to the volumes free of charge

Submission Manuscript

Only online submission, Web: https://journal.slaas.lk,e-ISSN: 2682-6992

Members of the Editorial Board

Editor in Chief

Prof. K P S Chandana Jayaratne
Department of Physics
University of Colombo,
Sri Lanka.
email: chandanajayaratne@gmail.com

Co- Editor
Dr. R Chinthaka L De Silva
Petroleum & Lubricant Testing Laboratory
Industrial Technology Institute
Sri Lanka.
363, Bauddhaloka Mw, Colombo, Sri Lanka.
email: chinthakades@yahoo.com

General President SLAAS 2026
Dr. Parakrama Weligamage
Agricultural Economics & Business
Management , University of Peradeniya
email: parakw@gmail.com

International Editorial Advisory Board
Prof. Don Nalin Nilusha Wijayawardene
Dr. Udara Abeysekara
Prof. Hemamala Karunadasa
Dr. M Wasim Siddiqui

Managing Editor
Dr. Lochandaka Ranathunga
Department of Information Technology
Faculty of Information Technology
University of Moratuwa, Sri Lanka
email: lochandaka@uom.lk

Editorial Board
Prof. Mahesh Jayaweera
Prof. B C Liyanage Athapattu
Dr. K.R.M. Chandrathilaka
Dr. S N C M Dias
Eng. (Dr.) Chandana Kulasuriya
Dr. Nadeesha Hettige
Dr. Sujeeva Wijesiri
Dr. K.G.U. R Kumarasinghe
Dr. W A S N Perera

Sectional Representatives
Dr. K.R.M. Chandrathilaka
Dr. S N C M Dias
Dr. Chandana Kulasuriya
Dr. Nadeesha Hettige
Dr. Sujeeva Wijesiri
Dr. K.G.U. R Kumarasinghe
Dr. W A S N Perera
Dr. K.R.N. Harshani

Exchanges: Please address all requests to the Secretary, ***Sri Lanka Association for the Advancement of Science, "Vidya Mandiraya" 120/10, Wijerama Mawatha Colombo 07, Sri Lanka***

ISSN 1391-0256 Copyright © 2026 by the Sri Lanka Association for the Advancement of Science, Sri Lanka. All rights reserve. e-ISSN: 2682-6992

Table of Contents

SOLVING DETERMINISTIC MULTI-OBJECTIVE ASSIGNMENT PROBLEMS USING ANT COLONY OPTIMIZATION ALGORITHM

C. P. S. Pathirana[1*] and W. B. Daundasekera[2]

1Department of Mathematics & Philosophy of Engineering, Faculty of Engineering Technology, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka
[2] Department of Mathematics, Faculty of Science, University of Peradeniya, Peradeniya, Sri Lanka

ABSTRACT

This study is proposing a metaheuristic algorithm known as Ant Colony Optimization Algorithm (ACOA) to solve deterministic Multi-Objective Assignment Problems (MOAPs). While single-objective assignment problems have only one objective to optimize, MOAPs have several conflicting objectives to optimize simultaneously, such as minimizing production time and maximizing the productivity. ACOA is well-suited to solve this type of problems to a certain degree of accuracy because of the adaptive search and the collaborative learning. In this study, the ACOA is applied to seek a feasible near optimal solution to the deterministic assignment model, where in the implementation of the algorithm, the artificial ants construct assignments based on pheromone trails and heuristic desirability derived from the probabilistic estimates. The mechanism of pheromone evaporation and probabilistic path selection permit the algorithm to locate the near optimal solution while adhering to a set of constraints. The proposed algorithm is coded in Python programming language and the efficiency of the algorithm in solving MOAPs is compared with an existing prominent method known as Technique for an Order of Preference by Similarity to Ideal Solution (TOPSIS). The proposed Multi-Objective Ant Colony Optimization Algorithm (MOACOA) is proven to be capable of solving large scale MOAPs in less computational time compared to the TOPSIS method. The extension of this study can focus on solving stochastic MOAPs.

Keywords: Ant colony Optimization, Assignment problem, Cost matrix, Multi-objective, Pheromone evaporation.

----

*cppat@ou.ac.lk

INTRODUCTION

Decision-making in real-world environments often involves allocating a set of limited resources to a set of tasks while simultaneously optimizing multiple objectives which are conflicting in nature. Production planning, personnel assignment, transportation, machine allocation, and project management are just a few of the many areas where this occurs. The classical Assignment Problem (AP) is a well-known optimization problem that determines the optimal one-to-one assignment of a set of agents (e.g., workers, machines, etc.) to an equal number of tasks, with the objective of minimizing total assignment cost or maximizing total profit, subject to the constraints that each agent is assigned to exactly one task and each task is performed by exactly one agent. However, in practical situations, decision-makers are rarely concerned with a single criterion. Multiple performance measures such as cost, time, efficiency, reliability, and risk must often be considered concurrently. In Multi-Objective Assignment Problem (MOAP), the goal is to determine the optimal assignment that achieves a balance among several conflicting objectives (Alaya et al, 2007). The computational time of solving MOAP increases with the problem size and the number of objectives, and traditional algorithms are computationally demanding to solve large scale problems. To address this challenge, metaheuristic algorithms have been employed due to their ability to obtain high-quality near-optimal solutions within reasonable computational time. In this study, a metaheuristic algorithm, named Ant Colony Optimization Algorithm (ACOA), was considered to obtain the optimal solution to the MOAP. Further, the study shows that the proposed algorithm is computationally efficient in solving the MOAP. The proposed approach seeks to identify near-optimal compromise solutions among a large number of feasible assignments, recognizing that exact optimality is often computationally impractical for large-scale MOAPs.

The ACOA was first proposed by Dorigo et al. to solve NP-hard optimization problems such as the Traveling Salesman Problem, and it has since been extended to address a wide range of discrete optimization problems [(Dorigo et al., 1991), (Dorigo et al., 1999), (Rekaya et al., 2013) and (Monteriro et al., 2012)]. Ant Colony Optimization (ACO) is a metaheuristic introduced by Marco Dorigo, inspired by the collective foraging behavior of biological ants (Dorigo et al., 1991). The algorithm was originally developed in 1992 as part of Dorigo's PhD research [(Dorigo et al., 1991) and (Dorigo, 1992)]. ACO is based on the natural process by which ants search for shortest paths between their colony and food sources. The field of ant colony optimization, which is a biologically inspired method, has been shaped by a number of researchers who have made valuable contributions to the development and success of the field. The field of ''ant colony algorithms'' studies models derived from the observation of real ants' behavior and uses these models as a source of inspiration for the design of novel algorithms to solve optimization and distributed control problems [(Dorigo et al., 1991), (Dorigo, 1992) and (Alizaden et al, 2014)]. The ACOA is a probabilistic technique which can be used to find the optimal solution of the AP. Recent studies [(Blum, 2005), (Gulben, 2015) and (Rekaya, 2013)] have demonstrated that the Ant Colony Optimization Algorithm (ACOA) performs competitively compared to exact methods and other metaheuristic approaches when solving large-scale assignment problems. Although exact methods guarantee optimal solutions, they often become computationally impractical for high-dimensional problems. In contrast, ACOA is capable of producing high-quality near-optimal solutions within a significantly less computational time, making it a practical and efficient approach for real-world task scheduling problems subject to operational constraints.

The first extensions of ACOA for Multi-Objective Ant Colony Optimization Algorithm (MOACOA) were proposed at the end of the 1990s. Its initial application area was single objective AP. However, the existing ACOA was extended to solve MOAP, where the quality of candidate solutions is evaluated according to several conflicting objectives. One of the fundamental challenges in Multi-Objective Ant Colony Optimization (MOACO) is the representation and management of pheromone information for multiple objectives. Two main strategies have been proposed to overcome the challenge: use of a single pheromone matrix and use of multiple pheromone matrices (Maniezzo et al., 2010). When a single pheromone matrix is employed, pheromone updates are typically performed using some or all non-dominated solutions obtained during the search (Iredi et al., 2001). In contrast, approaches based on multiple pheromone matrices, generally assign a separate pheromone structure to each objective, with pheromone updates guided by elite solutions corresponding to the individual objectives (Alaya et al., 2007; Doerner et al., 2004). During the solution construction process, information from multiple pheromone and heuristic matrices is usually combined through weighting schemes to balance the influence of the different objectives (Maniezzo et al., 2010). Although the behavior of a single ant is relatively simple, the cooperative interaction among many ants gives rise to an intelligent and adaptive search mechanism capable of solving complex optimization problems (Syariza et al., 2018). This self-organizing behavior has motivated the development of ACO-based algorithms for a wide range of combinatorial and multi-objective optimization problems, including assignment, scheduling, routing, and resource allocation.

An ant colony is highly organized, in which one ant interacts with others through pheromone in perfect harmony. While perhaps seemingly simple, this is indeed a very complex behavior (White, 1984). The complex behavior of ants has inspired researchers to find solutions to certain optimization problems. The ACOA is developed based on biological behavior of ants seeking a path between their colony and source food (Pathirana et al., 2024). In this study, Multi-Objective Ant Colony Optimization Algorithm (MOACOA) is proposed which is a probabilistic approach for finding the optimal solution of the MOAP. The optimal assignment found by applying this proposed method is verified by applying an existing method known as Technique for an Order of Preference by Similarity to Ideal Solution (TOPSIS) (Hwang et al. 1981; Starkweather, 1991).

The TOPSIS method was first proposed by Hwang and Yoon as a distance-based Multi-Criteria Decision-Making (MCDM) approach. TOPSIS is particularly useful in problems where alternatives must be ranked or selected based on multiple criteria, such as assignment problems, resource allocation, and decision optimization (Hwang et al. 1981). The fundamental principle of TOPSIS is that the chosen alternative should have the shortest distance from the positive ideal solution (PIS) and the farthest distance from the negative ideal solution (NIS). The PIS represents the best possible performance across all criteria, while the NIS represents the worst. Since its introduction, TOPSIS has become one of the most widely used methods. In multi-objective assignment problems (MOAP), TOPSIS is often used as a decision-support tool to select the most preferred assignment from a set of Pareto-optimal or feasible solutions generated by optimization algorithms (Behzadian et al., 2012). These strengths make TOPSIS particularly suitable for evaluating solutions obtained from algorithms such as genetic algorithms, ant colony optimization, and other metaheuristic approaches. In the context of the present research, TOPSIS is employed as a multi-criteria decision-making framework to evaluate and select optimal or near-optimal assignment solutions. TOPSIS uses as a suitable decision-support technique for solving multi-objective assignment problems. In this research, computational results demonstrate that the proposed algorithm is capable of solving large-scale MOAPs with significantly less computational time than the TOPSIS method.

MATERIALS AND METHODS

The general formulation of the mathematical model of the MOAP can be represented as below.
Parameters:

m	number of objectives
n	number of agents/number of tasks
c_{ij}^k	cost of assigning j^{th} task to i^{th} agent for k^{th} objective

$$\text{Min } Z^k(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^k x_{ij} \quad ; \; k = 1,2,\ldots,m$$

subject to the constraints

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad ; i = 1, 2, \dots, n \quad \text{- Agent assigned to a task}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad ; j = 1, 2, \dots, n \quad \text{- Task assigned to an agent,}$$

where x_{ij} is defined as

$$x_{ij} = \begin{cases} 1, & \text{if } i^{th} \text{agentis assigned to } j^{th} \text{task,} \\ 0, & \text{if } i^{th} \text{ agent is not assigned to } j^{th} \text{ task} \end{cases}$$

Furthermore, the standard matrix of the multi-objective assignment problem is as follows:

$$\left[c_{ij}^{k}\right]_{n\times n} = \begin{pmatrix} c_{11}^{k} & c_{12}^{k} & \cdots & c_{1n}^{k} \\ c_{21}^{k} & c_{22}^{k} & \cdots & c_{2n}^{k} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1}^{k} & c_{n2}^{k} & \cdots & c_{nn}^{k} \end{pmatrix}_{n\times n} \quad ; k = 1, 2, \dots, m.$$

In this model all parameters are assumed to be deterministic.

Considering the proposed method of constructing the probabilistic solution for the ant colony system, when using multiple matrices, where each matrix is corresponding to a specific objective, the matrices are added together using weighted sum, where weights are assigned according to the priorities given to the objectives. In this study, a linear aggregation approach is adopted, as the proposed ACOA ensures that all pheromone values are maintained within the same range. Within the ACOA framework, the probabilistic solution construction mechanism involves assigning an initial pheromone value and estimating the selection probability for each candidate solution. In the proposed method, the procedure begins with the first row of each Assignment Cost Matrix (ACM).

Initially, transition probabilities (p_{ij}^{q}) of all paths in the first row of the linear aggregation of ACMs are calculated. Based on these probabilities, the cumulative probabilities (CP) for that row will also be calculated. For the random selection of the new path, a random number between 0 and 1 is generated uniformly for each selected path. Comparing the CP and the generated random number the new path is determined. If the comparison between the CP and the generated random number across all candidate paths does not yield a unique path, that is, if multiple paths become candidates, the pheromone levels are updated and the transition probabilities are recomputed accordingly, and then the process described earlier will be repeated. This procedure is repeated for the corresponding row of the linearly aggregated ACMs until the comparison converges to a unique shortest path. After selecting the shortest path for the corresponding row, this process is repeated sequentially for the remaining rows of the ACMs as well.

The transition probability of the q^{th} ant moving from agent i to task j is as follows (Gulben, 2015; Manuel et al. 2010)]:

$$p_{ij}^{q} = \frac{(\sum_{l=1}^{k} w_l \tau_{ij}^{l})^{\alpha} (\sum_{l=1}^{k} w_l \eta_{ij}^{l})^{\beta}}{\sum_{j\in N_i^q}[(\sum_{l=1}^{k} w_l \tau_{ij}^{l})^{\alpha} (\sum_{l=1}^{k} w_l \eta_{ij}^{l})^{\beta}]},$$

where N_i^q is the feasible neighborhood of ant q and $\eta_{ij} = 1/c_{ij}$, where c_{ij} is the cost between task *j* and agent *i*. Also τ_{ij}^{l} is the pheromone trails for arc (i, j) for the objective, η_{ij}^{l} is the corresponding heuristic value, w_l is a weight rule thus takes the form:

$$\tau_{ij}^{k}(t+1) \leftarrow (1-\rho) w_k \tau_{ij}^{k}(t) + \sum_{a=1}^{r} \Delta\tau_{ij}^{a},$$

where ρ is the evaporation rate of pheromone, $\rho \in [0,1]$, r is the number of paths of ants, $\tau_{ij}(t+1)$ is the total pheromone deposit by ant on path (i,j) and $\Delta\tau_{ij}^{a}$ is the quantity of pheromone laid on the edge (i,j) by ant a. In ant colonies, pheromone intensity decreases over time because of evaporation. In MOACOA, evaporation is simulated by applying an appropriately defined pheromone evaporation rule:

$$\Delta\tau_{ij}^{a} = \begin{cases} \dfrac{1}{c_a}, & a^{th}\text{ant's assignment cost on the edge } (i,j), \\ 0, & \text{otherwise.} \end{cases}$$

This proposes the following process, depicted in the step-wise form, to obtain optimal assignment for an MOAP:

Step 1: Formulate the Assignment Cost Matrix (ACM) for each objective.

Step 2: Define the values of α, β, ρ, w_l and initial pheromone level (τ_{ij}^{0}).

Step 3: Determine the transition probability values (p_{ij}^{q}) for the first row of the linear aggregation ACM.

Step 4: Compute the CP and generate the RNs in the corresponding row.

Step 5: Find the shortest path for the first row of the linear aggregation ACM comparing the CP and RNs.

Step 6: Exclude the rows and columns of ACMs corresponding to the selected shortest path.

Step 7: If the shortest path cannot be found, update the pheromones and go to Step 3.

Step 8: If the shortest path is found, repeat the process for the remaining rows of ACMs.

A flowchart of the proposed algorithm is given in the Figure 1 below:

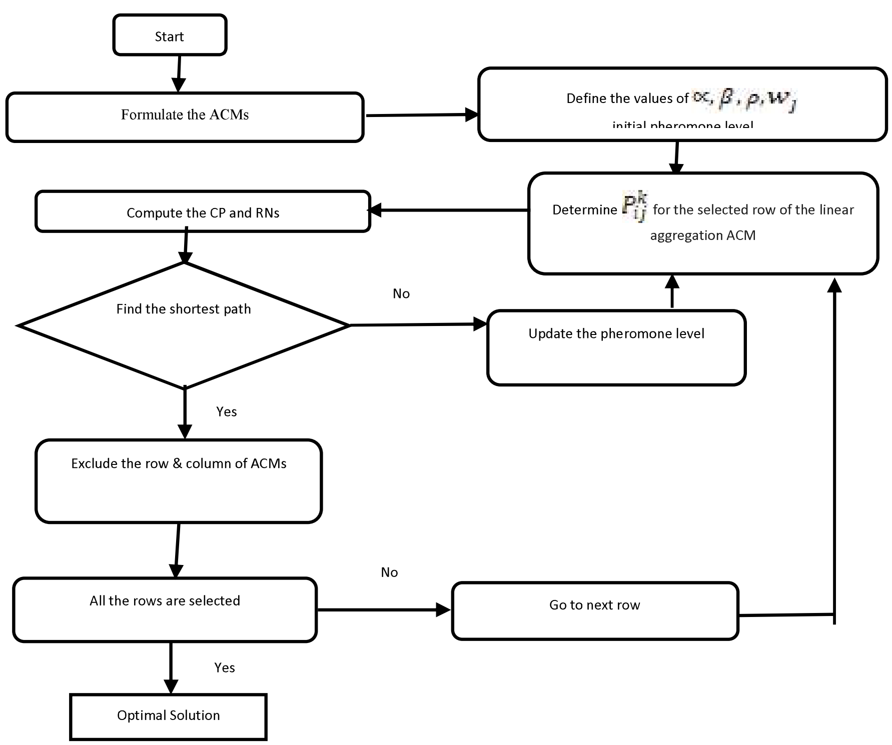

Figure 1. Flowchart of the proposed algorithm

The optimal solution obtained for the MOAP is evaluated using the TOPSIS method, which consists of the following steps:

Step 1: Formulate the Assignment Cost Matrix (ACM) for each objective.

Step 2: Define the values of w_j.

Step 3: Normalize the decision matrix for each objective.

Step 4: Compute the weighted normalized matrix.

Step 5: Determine the PIS and NIS.

Step 6: Calculate the Separation from the PIS and NIS for each alternative path.

Step 7: Compute the closeness coefficient for each alternative path.

Step 8: Rank the alternatives according to the closeness coefficient.

Step 9: Select the alternative with the highest rank as the best solution.

A flowchart of the TOPSIS method is given in the Figure 2 below:

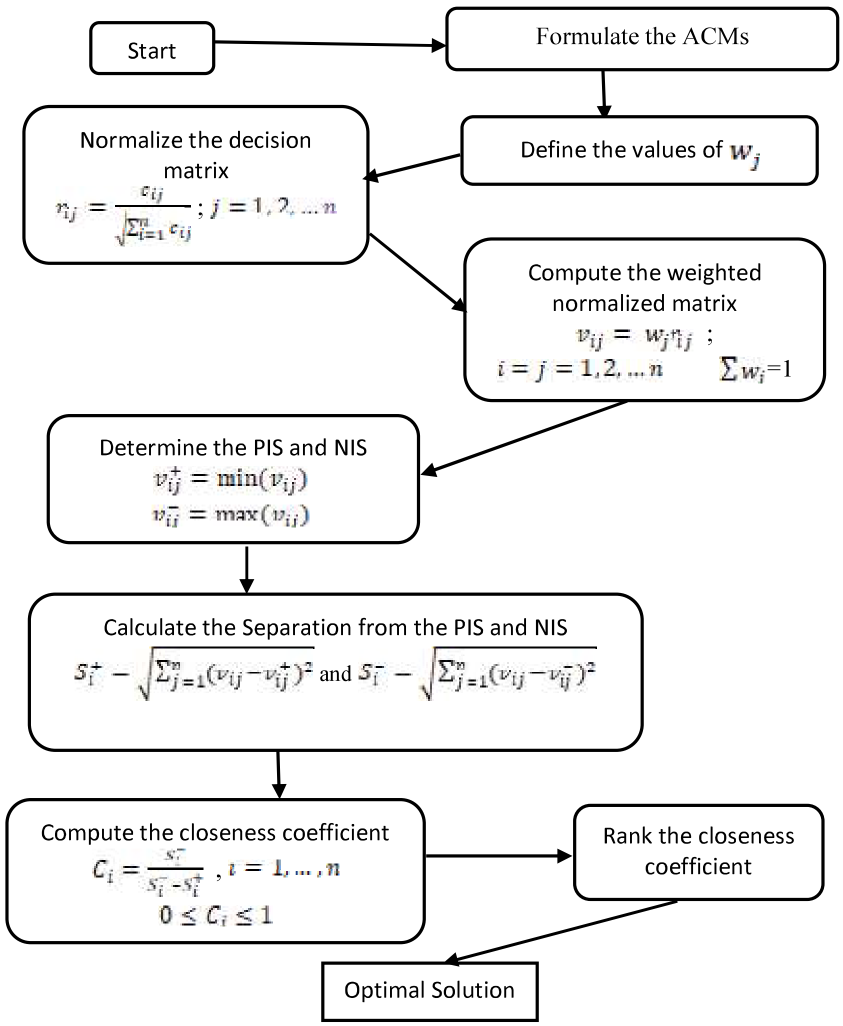

Figure 2. Flowchart of the TOPSIS method

RESULTS AND DISCUSSION

Illustration 1 with two objectives

This illustration, which consists of two objectives, considers four agents to be assigned for four tasks in the optimal manner, where assignment costs c_{ij} are randomly generated for the illustrative purpose, are given below in the matrix form, where c^1, c^2 are ACMs for the two objectives respectively:

$$c^1 = \begin{pmatrix} 77 & 6 & 62 & 7 \\ 39 & 72 & 95 & 59 \\ 34 & 82 & 27 & 6 \\ 75 & 42 & 27 & 77 \end{pmatrix} \quad \text{and} \quad c^2 = \begin{pmatrix} 36 & 55 & 1 & 95 \\ 8 & 23 & 78 & 20 \\ 42 & 3 & 34 & 20 \\ 14 & 55 & 78 & 68 \end{pmatrix}$$

The MOAP can be formulated as below according to the notations defined earlier:

$$Min\, Z^k(x) = \sum_{i=1}^{4} \sum_{j=1}^{4} c_{ij}^k x_{ij}, \;\; k = 1,2.$$

subject to the constraints:

$$\sum_{j=1}^{4} x_{ij} = 1, \qquad i = 1,2,3,4$$

$$\sum_{i=1}^{4} x_{ij} = 1, \qquad j = 1,2,3,4$$

$$x_{ij} = \begin{cases} 1, & \text{if } i^{th} \text{resource assign to } j^{th} \text{task} \\ 0, & \text{if } i^{th} \text{ resource not assign to } j^{th} \text{ task} \end{cases}$$

Since two objectives are considered, the transition probability can be rewritten as follows (Let $\alpha, \beta = 1$):

$$p_{ij}^k = \frac{(w_1\tau_{ij}^1 + w_2\tau_{ij}^2)(w_1\eta_{ij}^1 + w_2\eta_{ij}^2)}{\sum(w_1\tau_{ij}^1 + w_2\tau_{ij}^2)(w_1\eta_{ij}^1 + w_2\eta_{ij}^2)}$$

Without loss of generality we take $w_1 = 0.5$ and $w_2 = 0.5$ ($\sum_{i=1}^{2} w_i = 1$).

According to the proposed method given in the step-wise form, initially the first row of each of the ACMs is considered. Subsequently, the η_{ij}^1, η_{ij}^2 and transition probabilities for the paths are calculated in the first row of each ACM as shown in Table 1 below. The model is solved using ACOA coded in Python program and without loss of generality, let the initial pheromone value to be 1:

Table 1: Transition probability on the first rows of the ACMs

Path	1st row (c^1)	1st row (c^2)	τ_{ij}^1	τ_{ij}^2	η_{ij}^1	η_{ij}^2	P_{ij}^1
$1 \to 1$	77	36	1	1	0.0130	0.0278	0.0292

$1 \to 2$	6	55	1	1	0.1667	0.0182	0.1325
$1 \to 3$	62	1	1	1	0.0161	1.0000	0.7283
$1 \to 4$	7	95	1	1	0.1428	0.0105	0.1099

Initially, the cumulative probabilities are computed, and the corresponding RNs are generated for the first row of the matrix obtained by the linear aggregation of the ACMs. Comparing the CP and the generated RNs, for example, if the random number 0.0619 is generated, the new path from $1 \to 2$ is selected by the ant, because the corresponding transition probability falls in the interval (0.0292, 0.1617), as shown in the column 4 of Table 2. Similarly, new paths are selected by comparing the random numbers and CP shown in the remaining columns in Table 2 below:

Table 2: Trasition probabilities and new paths of first rows of the respective ACM

Path	$P^1_{i,j}$	Cumalative Probability	Path $1 \to 1$	Path $1 \to 2$	Path $1 \to 3$	Path $1 \to 4$
$1 \to 1$	0.0292	0.0292	RN: 0.0619 and new path $1 \to 2$	RN: 0.1889 and new path $1 \to 3$	RN: 0.7255 and new path $1 \to 3$	RN: 0.3889 and new path $1 \to 3$
$1 \to 2$	0.1325	0.1617				
$1 \to 3$	0.7283	0.8900				
$1 \to 4$	0.1099	1.0000				

As shown in Table 2, the paths $1 \to 2$, $1 \to 3$, and $1 \to 4$ converge to the same path, that is $1 \to 3$, while the remaining path follows the path $1 \to 2$. In the subsequent step, the pheromone values (total pheromene) are updated for the newly selected paths, as presented in Table 3. During the second phase, the pheromone evaporation rate (ρ) is set to 0.5.

Table 3: τ^1_{ij} , τ^2_{ij}, η^1_{ij} and η^2_{ij} values for the first row in the second phase

Path	τ^1_{ij}	τ^2_{ij}	c^1_{ij}	η^1_{ij}	c^2_{ij}	η^2_{ij}
$1 \to 2$	0.2630	0.2778	6	0.1667	55	0.0182
$1 \to 3$	0.5756	1.2787	62	0.0162	1	1

The new transition probabilities for paths in the first row of each ACM in the second phase are calculated, and the results obtained are shown in Table 4 below:

Table 4: Paths for the first row in the second phase

Path	$P_{i,j}^2$	Cumulative Probability	Path $1 \rightarrow 1$	Path $1 \rightarrow 2$
$1 \rightarrow 2$	0.0504	0.0504	RN: 0.7630 and new path $1 \rightarrow 3$	RN: 0.7839 and new path $1 \rightarrow 3$
$1 \rightarrow 3$	0.9496	1.0000		

Subsequently, comparing the CP and the generated RNs as before, the path for the ants is selected. That is, the same path from $1 \rightarrow 3$ is taken for both paths $1 \rightarrow 2$ and $1 \rightarrow 3$. Therefore, the optimal assignment for the first row of the corresponding ACMs is determined as $1 \rightarrow 3$. The optimal solution for the first row of each ACMs is recorded. Subsequently, the row and column of each ACMs corresponding to the selected path are excluded. This procedure is then repeated for the remaining rows, and the optimal solution corresponding to each row is recorded. Finally, the proposed iterative technique converges to the optimal assignment for the MOAP.

The optimal assignment is {(1, 3), (2, 1), (3,4), (4, 2)} and the total assignment cost is calculated as:

$$\sum_{i=1}^{4}\sum_{j=1}^{4} c_{ij}^k x_{ij} = (62 + 1) + (39 + 8) + (6 + 20) + (42 + 55) = 63 + 47 + 26 + 97 = 233.$$

The TOPSIS method is an important multi-criteria decision-making (MCDM) approach for solving MOAP. It is one of the most widely used and conceptually simple method. The solution obtained using the TOPSIS method is

{{0, 0, 1, 0}, {1, 0, 0, 0}, {0, 0, 0, 1}, {0, 1, 0, 0}}.

From this result, the optimal assignment is {(1,3),(2,1),(3,4),(4,2)}, and the total assignment cost is 149 + 84=233. Hence, the result obtained by the proposed algorithm for the two objectives is verified using the TOPSIS method.

Illustration 2 with three objectives

This illustration, which consists of three objectives, considers four workers to be assigned for four jobs in the optimal manner, where assignment costs c_{ij} are randomly generated for the illustrative purpose, and are given below in the matrix form, where c^1, c^2, c^3 are ACMs for the three objectives respectively:

$$c^1 = \begin{pmatrix} 2 & 5 & 4 & 7 \\ 3 & 3 & 5 & 7 \\ 3 & 8 & 4 & 2 \\ 6 & 5 & 2 & 5 \end{pmatrix}, c^2 = \begin{pmatrix} 3 & 3 & 6 & 2 \\ 5 & 3 & 7 & 2 \\ 5 & 2 & 7 & 4 \\ 4 & 6 & 3 & 5 \end{pmatrix} \text{ and } c^3 = \begin{pmatrix} 4 & 2 & 5 & 3 \\ 5 & 3 & 4 & 3 \\ 4 & 3 & 5 & 2 \\ 6 & 4 & 7 & 3 \end{pmatrix}$$

Assignment model with three objectives

$$Z^k(x) = \sum_{i=1}^{4}\sum_{j=1}^{4} c_{ij}^k x_{ij}, \quad k = 1,2,3$$

subject to the constraints

$$\sum_{j=1}^{4} x_{ij} - 1, \qquad i - 1,2,3,4$$

$$\sum_{i=1}^{4} x_{ij} = 1, \qquad j = 1,2,3,4$$

$$x_{ij} = \begin{cases} 1, & \text{if } i^{th} \text{resource is assigned to } j^{th} \text{task} \\ 0, & \text{if } i^{th} \text{ resource is not assigned to } j^{th} \text{ task.} \end{cases}$$

Without loss of generality we take $w_1 = 0.4$, $w_2 = 0.3$ and $w_3 = 0.3$ ($\sum_{i=1}^{3} w_i = 1$). Here, using the MOACOA coded in Python programming language to solve the illustrative example with three objectives, the proposed iterative technique successfully converges to the optimal assignment of the MOAP. The optimal assignment is
{(1, 1), (2, 2), (3, 4), (4, 3)} and the total assignment cost is calculated as,

$$\sum_{i=1}^{4}\sum_{j=1}^{4} c_{ij}^{k} x_{ij} = (2 + 3 + 4) + (3 + 3 + 3) +(2 + 4 + 2) + (2 + 3 + 7) = 38.$$

As before, the optimal assignment obtained through the steps of the TOPSIS method is
{{1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 0, 1}, {0, 0, 1, 0}}.
Based on this result, the corresponding optimal assignment is {(1, 1), (2, 2), (3, 4), (4, 3)}, yielding a total assignment cost of 9+ 9 + 8 + 12 = 38. Hence, the results obtained by the proposed algorithm for the illustration with three objectives is verified by the TOPSIS method.

Comparision of computational efficency between MOACOA and TOPSIS method in solving the MOAP

The proposed algorithm is coded in Python programming language and is verified using the TOPSIS method. The TOPSIS method is a well known traditional algorithm to solve MOAPs. This method attempts to reach the optimal solution in contrary to the MOACOA. In this study, the performances of the TOPSIS method and the MOACOA are evaluated in terms of computational time required to reach an optimal solution for the MOAP, by varying the matrix size from 2 to 10 for two and three objective functions. Table 5, which is given below, exhibits the results of the analysis. The last row of the table shows that the TOPSIS method fails to reach the optimal solution in both cases where the matrix size exceeds 11.

Table 5: A comparison of computational time between the MOACOA and TOPSIS method to solve MOAP

Size of ACM	Elapsed time (Seconds)			
	Two objectives		Three objectives	
	MOACOA	TOPSIS Method	MOACOA	TOPSIS Method
2	0.0339	1.2744	0.0479	1.3439

3	0.0663	1.2958	0.0635	2.0112
4	0.1371	1.4598	0.1442	2.2113
5	0.1930	1.4631	0.1995	3.3480
6	0.2400	2.4995	0.3305	4.4108
7	0.4499	2.8681	0.4086	4.4949
8	0.5079	4.3824	0.5861	6.2688
9	0.5123	5.3954	0.6224	8.9179
10	0.7101	34.9261	0.7682	47.0873
11	0.7346	-	0.7698	-

According to the Table 5 given above, the TOPSIS method requires significantly more computational time to solve the problem as the size of the problem increases. For example, when the order of the ACM is 10, the TOPSIS method takes as long as 34.9261 and 47.0873 seconds respectively to obtain the optimal solution for the two cases, whereas MOACOA produces the near optimal solution in only 0.7101 and 0.7682 seconds respectively. The Figure 3 below clearly illustrates the computational time required by each method in both cases:

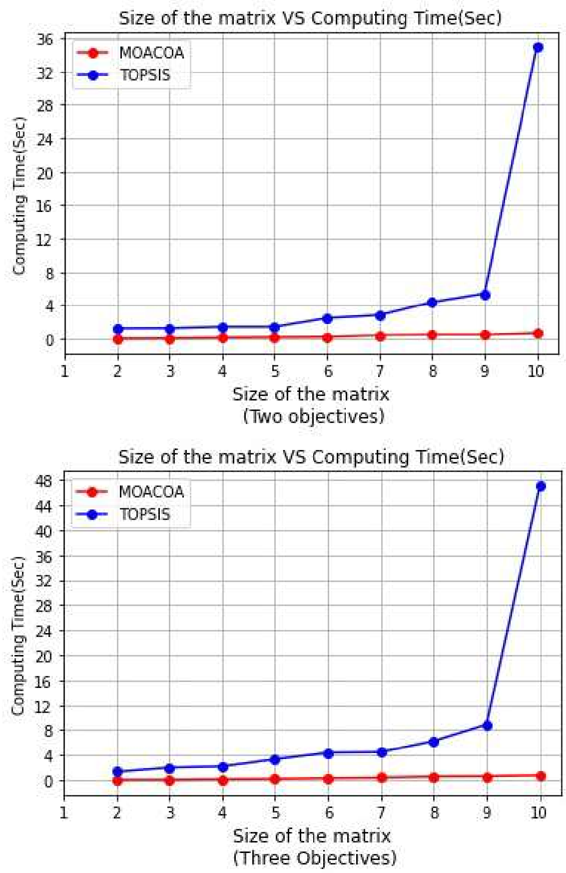

Figure 3. Size of the ACM vs Computational time (seconds) for two and three objectives

CONCLUSION

This study proposes a Multi-Objective Ant Colony Optimization Algorithm (MOACOA) for solving deterministic Multi-Objective Assignment Problems (MOAPs). The proposed algorithm, MOACOA, is coded in Python programming language and is verified its performances using the TOPSIS method. This proposed metaheuristic algorithm is capable of solving large scale MOAPs in less computational time compared to time taken by the TOPSIS method for the same. Table 5 shows that both algorithms provide identical assignments for small scale problems and MOACOA produced assignment solutions identical to those obtained by the TOPSIS method, confirming the accuracy and reliability of the proposed approach. The study shows the efficiency and robustness of MOACOA for solving real world large scale MOAPs. Hence, in this study it shows that for the constrained MOAPs, MOACOA stands out as a viable and smart solution that combines the maximum output and feasibility, which is optimal in circumstances where the TOPSIS method fails to be useful. Hence, TOPSIS becomes computationally less efficient or less practical for large-scale problems. This study recommends that the proposed method can be used more efficiently and effectively in solving large scale MOAPs.

The primary significance of this study is in the creation of the biologically-inspired heuristic framework that utilizes the combination of effective construction of the solutions and learning processes based on the pheromones for solving constrained MOAPs. This algorithm represents a valuable alternative to classical multi-criteria decision-making approaches in situations where the problem becomes computationally demanding.

Even though the current research focuses on deterministic scenarios only, the developed approach paves way for future studies concerning multi-objective assignment problems that involve uncertainties. Future studies may extend the proposed MOACOA to fuzzy, stochastic, and other uncertain multi-objective assignment models and investigate hybridization with other metaheuristic techniques to further improve solution quality and computational performance.

REFERENCE

[1] Alaya. I., Solnon. C., Gh´edira. K. (2007), Ant colony optimization for multi-objective optimization problems. In: 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), Los Alamitos, CA, vol. 1, pp. 450–457. IEEE, Computer Society Press, Los Alamitos.

[2] Alizadeh, B., Asghar Gholamian. S.A., (2014), Application of Ant Colony Optimization Algorithm for Optimal Design of Squirrel Cage Induction Moto, Journal of Mechatronics, Electrical and Computer Technology, Vol. 4(13), pp. 1674-1690.

[3] Behzadian, M., Otaghsara, S. K., Yazdani, M., & Ignatius, J. (2012). A state-of-the-art survey of TOPSIS applications. Expert Systems with Applications, 39(17), 13051–13069.

[4] Blum, C. (2005). Ant colony optimization: Introduction and recent trends. Physics of Life Reviews, 2(4), 353–373.

[5] Doerner. K., Gutjahr. W. J., Hartl, R. F., Strauss. C., Stummer. C. (2004). Pareto ant colony optimization: A metaheuristic approach to multi-objective portfolio selection. Annals of Operations Research 131, 79–99.

[6] Dorigo. M., Maniezzo,V., and Colorni, A. (1991).“Positive Feedback as a Search Strategy”, Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy.

[7] Dorigo. M., (1992). Optimization, learning and natural algorithms. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano.

[8] Dorigo. M, Di Caro. G, and Gambardella. L. M. (1999). Ant algorithm for discrete optimization, Artificial Life, MIT Press, 5137-72.

[9] Gulben. C, Orhan.Y,(2015), “An improved ant colony optimization algorithm for construction site layout problems”, Building construction and planning research, 3, 221-232.

[10] Hwang, Ching-lai and Yoon. (1981), Multiple Attribute Decision Making: Methods and Applications, Springer-Verlag, New York.

[11] Iredi. S., Merkle. D., Middendorf. M. (2001). Bi-criterion optimization with multi colony ant algorithms. In: Zitzler, E., Deb, K., Thiele, L., Coello, C.A., Corne, D.W. (eds.) EMO LNCS, vol. 1993, pp. 359–372. Springer, Heidelberg.

[12] Kantorovich. L., V. Etal. (1965), The best use of economic resources. The best use of economic resources.

[13] Manuel L, Thomas S. (2010), “An Analysis of Algorithmic Components for Multi-objective Ant Colony Optimization: A Case Study on the Bio-objective TSP”, Springer- Verlag Berlin Heidelberg, P. Collet et al. (Eds.): EA 2009, LNCS 5975, pp. 134–145.

[14] Monteriro. S. R., Dalila B. M. M. F., Fernando A. C. C. (2012). "Ant colony optimization: a literature survey", Economics and management, 474.

[15] Pathirana. C.P.S., Daundasekera. W.B, (2024), Applying Ant Colony Optimization Algorithm to Solve the Deterministic Single Objective Assignment Problem, Postgraduate Institute of Science (PGIS) Research Congress 2024, Volume 11, p58, 01st and 2nd November 2024.

[16] Rekaya. R., Robbins. K., Spangler. M., Smith. S., Hay. E. H., and Bertrand K. (2013). "Ant colony algorithm with applications in the field of genomics", Intech.

[17] Starkweather. T., McDaniel. S., Mathias. K, Whitley. D. (1991), A comparison of genetic sequencing operators, 4th International Conference on Genetic Algorithms (ICGA '91), University of California at San Diego, San Diego, pp 69-76.

[18] Syariza. S. F., Rahman. A and Rahman. R. A. (2018). "An Assignment Problem and Its Application In Education Domain: A Review and Potential Path", Hindawi, Advances in Operations Research, Volume, Article ID 8958393.

[19] White. D. J. A. (1984), special multi-objective assignment problem. Journal of the Operational research Society, 759–767.