Heliyon 10 (2024) e40134

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Application of an improved ant colony optimization algorithm of hybrid strategies using scheduling for patient management in hospitals

Md. Limonur Rahman Lingkon [a,*], Md. Sazol Ahmmed [a]

[a] Department of Industrial & Production Engineering, Rajshahi University of Engineering & Technology (RUET), Rajshahi- 6204, Bangladesh

ARTICLE INFO

Keywords:
Pheromone
Hybrid strategy
Diffusion mechanism
Assignment problem
Co-evolution

ABSTRACT

To balance the convergence speed and solution diversity and enhance optimization performance when addressing large-scale optimization problems, this research study presents an improved ant colony optimization (ICMPACO) technique. Its foundations include the co-evolution mechanism, the multi-population strategy, the pheromone diffusion mechanism, and the pheromone updating method. The suggested ICMPACO approach separates the ant population into elite and common categories and breaks the optimization problem into several sub-problems to boost the convergence rate and prevent slipping into the local optimum value. To increase optimization capacity, the pheromone update approach is applied. Ants emit pheromone at a certain spot, and that pheromone progressively spreads to a variety of nearby regions thanks to the pheromone diffusion process. Here, the real gate assignment issue and the travelling salesman problem (TSP) are chosen for the validation of the performance for the optimization of the ICMPACO algorithm. The experiment's findings demonstrate that the suggested ICMPACO method can successfully solve the gate assignment issue, find the optimal optimization value in resolving TSP, provide a better assignment outcome, and exhibit improved optimization ability and stability. The assigned efficiency is comparatively higher than earlier ones. With an assigned efficiency of 83.5 %, it can swiftly arrive at the ideal gate assignment outcome by assigning 132 patients to 20 gates of hospital testing rooms. To minimize the patient's overall hospital processing time, this algorithm was specifically employed with a better level of efficiency to create appropriate scheduling in the hospital.

1. Introduction

Dorigo introduced the ant colony optimization (ACO) method in 1992 [1]. It works as a population-based evolutionary algorithm, which is a heuristic that was encouraged by studies on the collective behavior of actual ants in the wild. It has been demonstrated that while tackling optimization issues, the performance of this ACO algorithm is comparatively far better than other algorithms. The ACO algorithm depends on the information-related feedback and the actions of several individuals. Even when ants only engage in very basic activities, the activity of the entire colony is acceptable. The ACO algorithm has three characteristics: distributed computing, positive feedback, and heuristic search. It is essentially an evolutionary algorithm's heuristic global optimization method [2]. The

* Corresponding author.
E-mail addresses: limonurrahman16@gmail.com (Md.L. Rahman Lingkon), ma8gf@mst.edu (Md.S. Ahmmed).

https://doi.org/10.1016/j.heliyon.2024.e40134
Received 12 March 2024; Received in revised form 10 August 2024; Accepted 4 November 2024
Available online 5 November 2024

information connection based on pheromones is crucial to the evolutionary process [3].

Owing to its benefits, the ACO method is frequently used to solve combinatorial optimization issues, Considering the issues with network routing, touring salespeople, assignments, job-shop scheduling, and vehicle routing, among others [4,5] Numerous professionals have dedicated their time to studying the ACO algorithm, and to tackle challenging optimization challenges, numerous enhanced ACO algorithms have been put forth [6,7]. A few improved outcomes and impacts have been acquired in the most recent years. However, when the complexity of the large-scale optimization issues increases, these enhanced ACO algorithms have several intrinsic flaws that make them difficult to solve, such as sluggish convergence speeds and local optimum values [8–10]. The multi-population strategy, co-evolution mechanism, pheromone update technique, and pheromone diffusion mechanism are incorporated to improve the optimization performance of the ACO algorithm [2,11]. The Ant Colony Algorithm was applied in various ways by a number of researchers to address various real-world issues in a variety of infrastructures [12]. It has been widely employed recently to solve shortest-path issues, as evidenced by several research publications [13]. Similarly, this method was employed for dynamic path planning in traffic congestion and other areas for research objectives [14]. As a result, we propose here a novel multi-population co-evolution approach (ICM-PACO) for ant colony optimization. The ICMPACO algorithm's ability to handle this particular hospital-related scheduling problem is tested using the travelling salesman problem (TSP) and the real gate assignment problem. With the use of data gathered from the hospital, the travelling salesman problem was solved by the algorithm, which produced a suitable timetable for the patients to follow. This improved efficiency would reduce the makespan, or overall processing time, at the hospital.

The rest of this paper is structured as follows: Related Works are presented in Section 2. The features of the Ant Colony algorithm are shown in Section 3. The features of Ant Colony Optimization for Co-Evolution of Multi-Population are explained in Section 4. The use of ICMPACO algorithms in the Travelling Salesman Problem (TSP) is shown in Section 5. Section-6 presents the sensitivity analysis. The discussion and restrictions are presented in Section 7. Finally, Section 8 presents the conclusion and suggestions for more study.

2. Related work

In recent years, several enhanced ACO algorithms have been presented by specialists and researchers to study the algorithm. Coelho suggested combining a differential evolution technique with a modified continuous approach of ACO [15]. Eight multi-objective ACO algorithms were put up by Rada-Vilela to solve the amount of time and assembly line of space balance challenge [16]. A multi-objective ACO method was suggested by him and Ma to investigate non-redundant linear sensor network design problems [7,17]. Juang et al. suggested a co-evolution continuous ACO method to solve design issues with accuracy-oriented fuzzy systems [7]. Swarm intelligence and local search are used in Yang et al.'s modified ACO algorithm to increase accuracy and efficiency [4,8]. Different deep-learning techniques have been applied for various goals in the recent past. Regression matched with the deep learning algorithm, for instance, was previously used to verify the impact on research output. These algorithms were also applied to improve the study of the drone picture [18]. Thus far, the simulation has been investigated in many real-world settings [19]. However, colonial algorithms continue to explore a few special areas, such as extreme image processing and aeroplane schedules [20]. Based on this research, A hybrid ACO algorithm based on traditional heuristic priority criteria combined with the ACO algorithm was presented by Myszkowski et al. [21]. A multi-objective ACO was presented by Ariyasingha and Fernando to address the majority of combinatorial optimization issues encountered in real life [22]. A co-evolutionary multi-ant colony optimization approach was presented by Jiang et al. for ship multi and branch pipe route construction under a variety of limitations [9,17]. For task-scheduling difficulties, Zuo et al. suggested a multi-objective optimization technique based on the ACO algorithm [7]. For the capacitated multiple-vehicle allocation of consumers, Bagherinejad and Dehghani introduced the non-dominated sorting ACO algorithm, a swarm intelligence-based method [22]. ACO system multi-pheromone variation proposed by Krynicki et al. [23]. For the 2D football simulation league, Chen and Wang presented a revolutionary attacking strategy based on multi-group ACO [24]. A multi-layer ACO algorithm was presented by Marzband for the real-time experimental implementation of the management system related to optimal energy [21,25]. Khan and Baig proposed a unique technique that uses the ACO minimum-redundancy-maximum-relevance to determine the relevant feature subset [26]. Modified ACO method, Zhang and Ai presented a unique ACO method for the multi-objective single-model assembly line balancing problem [27]. A number of innovative hybrid ant colony optimization-based methods were proposed by Huang and Yu to address the multi-objective job-shop scheduling issue with equal-size lot splitting [28–30]. Zhou et al. introduced a multi-objective multi--population ACO method for the continuous domain [10]. An innovative, resilient, and energy-efficient ACO routing algorithm was presented by Vijayalakshmi to improve the Max-Min-Path approach's performance [31]. Using the idea of the lazy ant, Tiwari and Vidyarthi presented an enhanced ACO algorithm that was auto-controlled [32]. For the generation of structural test data, Sharifipour suggested the ACO algorithm, which is based on evolution techniques [33]. Akka and Khaber proposed an improved ACO algorithm. It uses a stimulating probability to help the ant select the next grid and new heuristic information based on the concept of infinite step length to increase visibility accuracy and expand the vision field [34]. Chen and Shen proposed a unique population-based evolutionary optimization strategy, the elite-mixed ACO method, which is continuous with central initialization, to improve the accuracy of Takagi-Sugeno-Kang-type recurrent fuzzy network designs [35]. Other approaches have also been put out in recent years [7,35,36]. There were several research based on prediction of different diseases in different stages using the algorithms of machine learning. For example-optimization of recognition performance for the layers of the dropout was done using feature scaling by Ahmed Omar [37], prediction of Hepatitis C virus using a machine learning framework in Egypt was performed by Heba et al. [38], monkeypox skin lesions were done similarly using convolutional neural network (CNN) and urinary incontinence was predicted for female by Tarek et al. [39,40]. FDA-approved cancer drugs were found using synergistic combinations for machine learning algorithms developed by them also in another research work [41]. Optimization of the classifications for the diseases through the analysis of the model of symptoms of diseases was made in a research study [42].

The aforementioned examination of these associated works demonstrates the many ACO algorithms that the researchers have put forward, including auto-controlled ACO, memetic ACO, elite-mixed continuous ACO, multi-objective ACO, multi-group ACO, multi-layer ACO, and hybrid ACO algorithm. Better optimization outcomes are achieved by using these enhanced ACO algorithms to handle challenging optimization issues [43–45]. However, there is still a sluggish convergence speed and a high likelihood of falling into the local optimal value [46]. As a result, a more thorough investigation of an enhanced ACO algorithm with superior optimization performance is required. By incorporating the co-evolution mechanism, pheromone update strategy, pheromone diffusion mechanism, and multi-population strategy into the ACO algorithm, this work suggests a novel improved ant colony optimization (ICMPACO) technique [47].

3. ACO algorithm

Numerous repeats make up the ACO algorithm. In each cycle, several ants use heuristic data and the collective experience of previous ant populations to construct comprehensive solutions. The pheromone trail that remains on a solution's constituent elements, is used to symbolize these accumulated experiences [48–50]. Depending on the problem to be solved, the pheromone may be placed on the connections or the parts of a solution. The following is a description of the pheromone update rule technique-

3.1. Transition rule

In the ACO algorithm, a basic computational agent is an ant. It builds a solution to the current issue iteratively. Each ant advances from a state t to states, which correspond to a more comprehensive intermediate solution, during each algorithm iteration [28,51,52]. The following formula is used to choose the xth ant from state t to states from among the unvisited states that are memorized in P_t^x:

$$r = avg\max\left[\tau_i(t,u)^{\alpha}.\ \eta\ (t,u)^{\beta}\right], if\ q \leq q_0, u\epsilon p^x \tag{1}$$

The trial level is an assessment of that move's popularity made after the fact. When every ant has finished their solution, trails are often updated, corresponding to motions that were a part of "good" or "bad" solutions, the level of trails raising or lowering -

$$p_x(t,r) = \begin{cases} \dfrac{\tau(t,r)^{\alpha}.\ \eta\ (t,r)^{\beta}}{\sum\limits_{u\epsilon P_t^x} \tau(t,r)^{\alpha}.\ \eta\ (t,r)^{\beta}}, if\ r\ \epsilon\ P_t^x \\ 0 \quad otherwise \end{cases} \tag{2}$$

Here, p_x *(t, r)* represents the transition probability, $\tau(t,u)$ the pheromone concentration in the ith population between states t and u, $\eta\ (t,u)$ be the trail length between them, P_t^x the set of uninvited states of the population, α *and* β the control parameters and q work as a uniform probability having range [0, 1].

3.2. Pheromone update rule

The pheromone trails need to be updated in order to enhance the quality of the solution. Trail upgrading includes both local and global updates. The following is an explanation of the local trail update formula:

$$\tau(t,x) = (1-\rho)\tau(t,r) + \sum_{k=1}^{m} \Delta\tau_k(t,r) \tag{3}$$

In formula (3),

ρ $(0 < \rho < 1)$ = Evaporating rate of trial of pheromone,

$\Delta\tau_k(t,r)$ = Amount of trial of pheromone to the edge of (t,r) by ant k within interval t and (t+ Δt)

The description of the tour can be cleared by the following formula:

$$\Delta\tau_k(t,r) = \begin{cases} \dfrac{Q}{L_x} \quad if\ (t,x)\ \epsilon\ \pi_x \\ 0 \quad otherwise \end{cases} \tag{4}$$

In formula (4),

Q = Constant,

L_x = Distance to the sequence π_x toured for ant in time Δt.

4. Ant colony optimization for Co-Evolution of Multi-population

In real-world applications, the ACO algorithm solves optimization problems with better optimization performance, parallelism, and positive feedback. However, there are shortcomings such as poor convergence speed, complicated control parameter determination, premature convergence, and so on. The algorithm for co-evolution is a method for global optimization based on co-evolution phenomena found in the natural world. It embraces the concept of disintegration and cooperation to break down the intricate optimization and divide the issue into several interconnected optimization subproblems, which completely cooperate while each is optimized independently [53–55]. Thus, to propose a new multi-population co-evolution ant colony optimization (ICMPACO) algorithm for solving large-scale optimization problems in this paper, the multi-population strategy, the co-evolution mechanism, the pheromone updating strategy, and the pheromone diffusion mechanism are introduced into the ACO algorithm. The ICM-PACO algorithm uses the multi-population approach, which separates the ants into elite and common groups, to hasten convergence and prevent the ants from drifting toward the local optimum value. The optimization ability is enhanced by the pheromone update technique. Ants emit pheromone at a certain spot, and that pheromone progressively spreads to a variety of nearby regions thanks to the pheromone diffusion process. To execute information sharing, various subpopulations exchange information using the co-evolution method. The ACO algorithm's optimization performance is enhanced by the complete use of these tactics and processes.

4.1. Multi-population strategy

By controlling the solutions with the ant colony size, selection parameter, and convergence parameter, the ACO technique creates new solutions with a single species of ant. Finding the right parameter values to produce an improved ACO algorithm that avoids premature convergence and converges quickly is often challenging. Therefore, to enhance the performance of the ACO algorithm, a method related to multi-population is employed. Ants will be split into elite and common categories using this method. The elite ants can use the solution archive to gather knowledge, and they can use a probability selection approach together with a Gaussian kernel function to produce solutions. The elite ants are different in that they have their own set of rules. The elite ants vary in that they have their own set of rules. The elite ants are essentially employed to increase the ACO algorithm's pace of convergence.

To prevent themselves from reaching the local optimal value and to produce new solutions more slowly, common ants employ a single Gaussian function and the average value of each dimension. The following is a description of the normal ant's Gaussian function.

$$f_N^i(x) = \frac{1}{\sigma_{i,N}\sqrt{2\pi}} e^{-\frac{\left(x-\mu_{i,N}\right)^2}{2\sigma_{i,N}^2}} \tag{5}$$

$$\mu_{i,N} = \sum_{x=1}^{X} R_{i,x} \tag{6}$$

$$\sigma_{i,N} = \xi_N \sum_{l=1}^{X} \frac{\left(R_{i,e} - R_i\right)}{X-1} \tag{7}$$

In formula (5), (6) and (7),

$f_N^i(x)$ = Gaussian function of normal ants in ith dimension,

$\mu_{i,N}$ = Value of the sample,

$\sigma_{i,N}$ = Standard Deviation,

x_i = Average value for solutions of ith dimension,

ξ_N = Constant used for controlling convergence rate.

Consequently, common ants may successfully broaden their search area and improve their capacity for worldwide search.

4.2. Pheromone updating strategy

Pheromone updating, which includes both local and global pheromone updating, is one of the key problems with the ACO algorithm. To enhance the algorithm's efficiency in handling intricate optimization problems, a novel pheromone update technique and pheromone diffusion mechanism are suggested.

4.2.1. Local strategy

The pheromones on each edge are equal constants prior to the ACO algorithm's first iteration being run. On each passing edge for an ant in the ACO algorithm, the local pheromone update strategy is executed once any ant has finished the current iteration. The following is a description of the local pheromone update strategy's expression:

$$\tau_{a,b}^{(i)} = (1-\rho_L)\tau_{a,b}^{(i)} + \rho_L \Delta\tau_L^{(i)} \tag{8}$$

In formula (8),

$\rho_L \epsilon (0,1)$ = Pheromone (local) for the co-efficient,

$(1-\rho_L)$ = Reside factor of pheromone,

$\tau_0^{(i)}$ = The pheromone value (initial) has a small negative value when the node value equals 1 and 0 for the total value equals 0.

4.2.2. Global strategy

Following the completion of each ant's solution in the ACO algorithm, the passing nodes execute the global pheromone update strategy in a single iteration. The following describes the global pheromone strategy's expressions:

$$\tau_{a,b}^{(i)} = (1-\rho_G)\tau_{a,b}^{(i)} + \rho_G \Delta\tau_G^{(i)} \tag{9}$$

$$\tau_G^{(i)} = \begin{cases} F_G^{(i)}, (x,l) \in \mathit{Optimal\ solution}\ (\mathit{Global}) \\ F_I^{(i)}, (x,l) \in \mathit{Optimal\ Solution}\ (\mathit{Iterative}) \\ 0, \mathit{otherwise} \end{cases} \tag{10}$$

In formula (9) and (10),
$\rho_G\ \epsilon\ (0,1)$ = Evaporating Co-efficient of global pheromone.
$(1-\rho_G)$ = Residue factor of the pheromone.
$F_G^{(i)}$ = Optimal solution (Global).
$F_I^{(i)}$ = Optimal solution (Iterative).

4.2.3. Diffusion mechanism of the pheromone

The ant updates its pheromone using a single pheromone release mode. This option can only affect ants that have passed the same area in the future; it cannot guide the search for ants within a specific range of surrounding regions. It will thus affect optimization performance. This work uses the pheromone diffusion mechanism, which is based on either the local or global pheromone update approach, to improve the ACO algorithm. In general, there is a greater chance of a better answer in the nearby neighbourhood than in other areas. As a result, the pheromone generated by ants at a certain site may be made via the pheromone dispersion mechanism to progressively affect a specific range of nearby regions. When solving large-scale optimization problems, the other ants try to seek the nearby neighbourhood of the superior solution rather than the nearby neighbourhood of the inferior solution in an effort to improve optimization performance. The following is a description of the pheromone updating mechanism:

$$\tau_{a,b}^{(i)} = (1-\rho_E)\tau_{a,b}^{(i)} + \rho_L \Delta\tau_{a,b}^{(i)} \tag{11}$$

$$\tau_{a,b}^{(i)} = \begin{cases} \dfrac{1}{N+1} \times \dfrac{\tau_{a,b}^{(i)}}{d_r(O_a, O_b)}, d_r(O_a, O_b) \ < 1 \\ 0, \mathit{otherwise} \end{cases} \tag{12}$$

Begin
    Initialize
    While *stopping criterion and satisfied do*
        Position each ant in a satisfied do
        Repeat
            For each ant do
                Choose next node by applying the state transition rule
                Apply step by step pheromone update
            End for
        Until *every ant has built a solution*
        Update best solution
        Apply offline pheromone update
    End While
End

Fig. 1. Pseudo-code for the ant colony optimization (ACO).

In formula (10) and (11),

N = Obtained solution number in the iteration,

$\tau_a^{(i)}$ = Pheromone concentration left for guiding on the source object O_a,

$d_r(O_a, O_b) = 1/(f+1)$ is a distance of correlation between 2 objects.

4.3. Mechanism of Co-Evolution

A novel form of evolutionary mechanism based on co-evolution theory has emerged recently: the co-evolution mechanism. It acknowledges biological variety and places emphasis on the interdependence that exists during the evolutionary process between organisms and their surroundings. To increase optimization performance through the interaction of many populations, co-evolution theory is used to create the cooperation or competition connection among two or more populations. It highlights how several sub-populations now interact, influence one another, and coevolve together. Therefore, to achieve information contact among many sub-populations, the co-evolution mechanism is incorporated into the ACO algorithm.

4.4. ICMPACO algorithm model

Fig. 1 shows the ICMPACO algorithm model, which is based on the pheromone update strategy, pheromone diffusion mechanism,

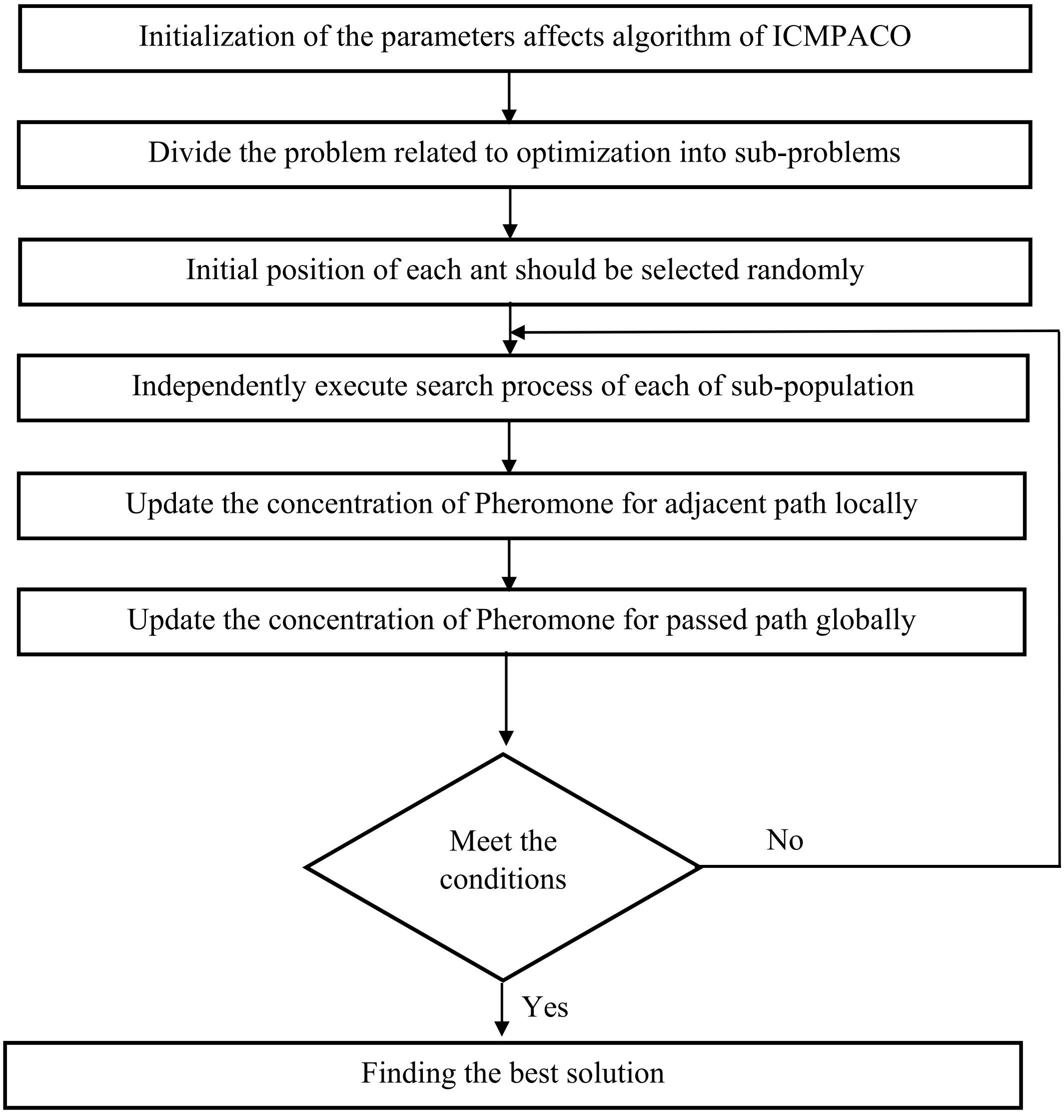

Fig. 2. Icmpaco algorithm model.

co-evolution mechanism, and hybrid strategy (see Fig. 2).

Step 1: Divide the optimization issue into several smaller problems, each relating to a single subpopulation.

Step 2: Set up the ICMPACO algorithm's parameters. Some examples of these parameters include the quantity of pheromone (Q), the number of ants (k), the maximum number of iterations (T), the volatility coefficient (ρ), the parameters (α and β), and so on.

Step 3: Choose each ant's starting location at random.

Step 4: Each subpopulation searches independently. Formula (2) is used to determine the following state's transition probability.

Step 5: Update the pheromone concentration of each subpopulation's traversed route locally using Formula (8).

Step 6: Based on the pheromone diffusion mechanism at Formulas (11) and (12), locally update the pheromone concentration of the surrounding route for each subpopulation.

Step 7: Using Formulas (8) and (10), update the global pheromone concentration for each route traveled.

Step 8: Proceed to the next step if every ant completes Step 4 ~ Step 7 in this iteration; if not, return to Step 4.

Step 9: Ascertain whether the obtained solution satisfies the criteria or if the maximum number of iterations (*T*) is reached. To begin a new evolution, do Step 4 if this last requirement is not met; if it is, move on to Step 10.

Step 10: The acquired solutions from each subpopulation are traded when ten iterations are finished in order to choose superior alternatives.

The proposed model architecture is provided below in Fig. 3.

Patients from various areas will arrive through the main door and proceed to the testing room, where they will utilize distinct gates to gain entry. The patient is free to wander about the room to do all of the necessary tests. Once all of the tests are performed, they will utilize EXIT to exit the hospital.

5. ICMPACO algorithms application to solve TSP

5.1. Travelling salesmen problem

Because the travelling salesman problem (TSP) is so simple to define but so challenging to solve, mathematicians and computer scientists, in particular, have paid close attention to it [54–56,57]. This issue may be summed up as follows: a hunt for the shortest closed tour that makes one and only one stop in each city. The representation of the TSP graph can be performed using a complete graph $X = (N, A)$, where, N is the vertices set which is called city also and A is the arc set, Distance $D = d_{ij}$ expresses the cost-related matrix connected to each of the arcs $(i,j) \in A$. In the equation of the cost matrix, D will be symmetric or may be asymmetric. The issue of finding the shortest closed tour that visits each of G is $n = |N|$ nodes precisely once is known as the TSP. In symmetric TSP, the distances between each pair of nodes are independent of the direction in which they traverse arcs, meaning that $d_{ij} = d_{ji}$ for each pair. We have $d_{ij} \neq d_{ji}$ in the asymmetric TSP, at least for one pair of nodes (i, j). Every TSP instance from the TSPLIB benchmark library that was utilized in the empirical research. The definition of the variables is expressed below:

$$d_{ij} = \begin{cases} 1, \textit{If arc } (i,j) \textit{ is included with the tour} \\ 0, \textit{otherwise} \end{cases} \tag{13}$$

The following generalization of a well-known integer program formulation can be used to formulate the TSP. The representation of

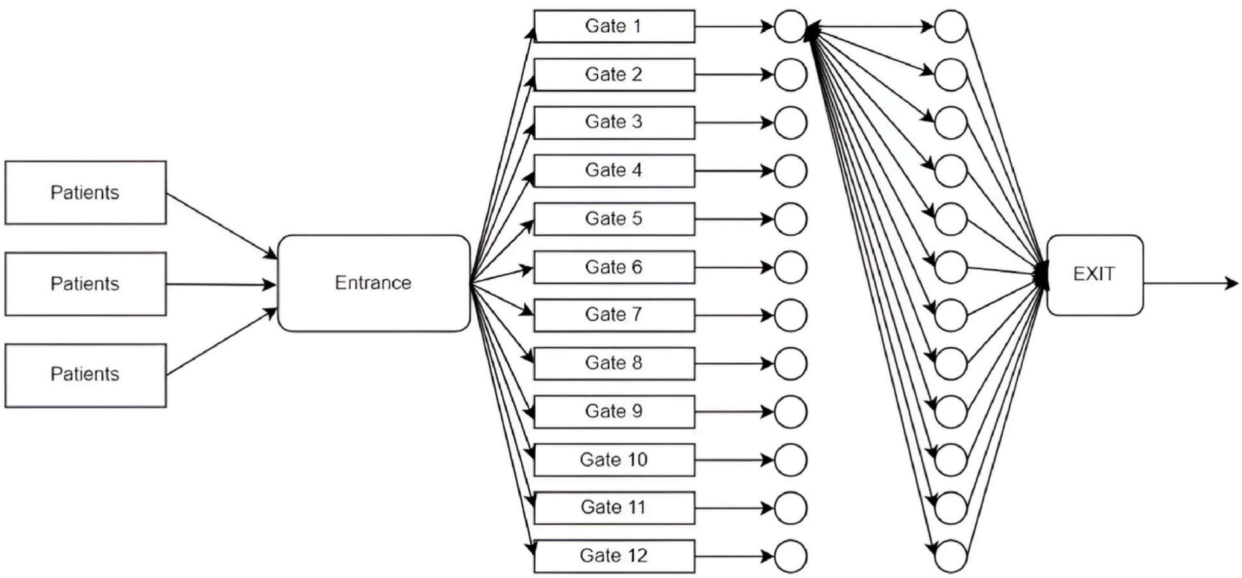

Fig. 3. Proposed model architecture.

the objective function is given below:

$$z = min\sum_{i}\sum_{j}^{x_{ij}d_{ij}} \tag{14}$$

The constraints for the problem are expressed as follows:

$$\sum_{i=1}^{n} d_{ij} = 1, j = 1, 2, 3, \ldots, n \tag{15}$$

$$\sum_{j=1}^{n} d_{ij} = 1, j = 1, 2, 3, \ldots, n \tag{16}$$

$$d_{ij} \epsilon \{0, 1\}, i, j = 1, 2, 3, \ldots, n \tag{17}$$

$$\sum_{i,j \epsilon S}^{n} d_{ij} \leq (S - 1), 2 \leq S \leq N - 2 \tag{18}$$

The total cost that has to be minimized is represented by the objective function (14) in these formulations. While constraint (16) ensures that every city is allocated to precisely one place, constraint (15) ensures that every position *j* is inhabited by a single city. The integrity constraints of variables zero–one x_{ij} ($x_{ij} \geq 0$) are represented by constraint (17). The final route assurances that no sub-routes would be created and that every city will only be visited once are provided by constraint (18).

5.2. Experimental setup including parameters

To demonstrate the optimization performance of the suggested ICMPACO approach, this research takes eight TSP standard examples from the TSPLIB standard library (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/). The distance between any two cities is calculated using the Euclidian distance and rounded off after the decimal point, according to the TSPLIB functionality. The optimization efficiency of the suggested ICMPACO method is compared with the standard ACO algorithm and the enhanced ACO (IACO) algorithm, which is based on swarm intelligence employing local search. The Pentium CPU CORE i7, 8.0 GB RAM, Windows 10, and Matlab2014b are the environments used for the experiments. Data was taken from a renowned private hospital in Bangladesh for 157 patients. Different primary data was taken from them and also from the authority of that hospital. Changes to the parameters in these algorithms may have an impact on the optimal value, making the parameter values themselves potentially complex problems. To determine the most logical starting values for these parameters, the alternative values were examined and adjusted for a few functions in the simulation experiments. To effectively solve the problem, these chosen parameter values take into account the best possible answer and the most acceptable running time of these algorithms. Therefore, via testing and modification, the most sensible beginning values of these parameters are found. Table 1 displays the starting values of these parameters that were acquired.

5.3. Experimental result and analysis

The proposed ICMPACO method, the IACO algorithm, and the basic ACO algorithm are all run ten times for every TSP standard instance in the simulation experiment. The experiment outcomes are described and contrasted using the indices of maximum value, minimum value, average value, and variation of the ten results. Table 2 displays the outcomes of the experiment. Here, the best value that was found is represented by the optimum value. The minimum value represents the achieved minimum value in the ten simulation tests, the maximum value is the highest value obtained, and the average value represents the average value of the ten simulation tests. The difference between the highest and minimum values is represented by the variance.

The experimental results for the IACO algorithm, the basic ACO technique, and the proposed ICMPACO algorithm in solving TSP standard cases of patients 1, 2, 3, 4, 5, 6, 7, and 8 are shown in Table 2. This table clearly illustrates the optimization and influence on the TSP solution of the fundamental ACO algorithm, the IACO technique, and the ICMPACO algorithm. The experiment findings for three optimization algorithms demonstrate that, in solving all TSP standard cases, the ICMPACO method can get the best optimization

Table 1
Parameter setup.

Parameters for Algorithms	ACO	IACO	ICMPACO
No. of Ants *(k)*	30	30	30
The factor for Pheromone (α)	01	01	01
The factor for Heuristics (β)	05	05	05
Co-efficient of Volatility (ρ)	0.1	0.1	0.1
Amount of Pheromone *(Q)*	100	100	100
Initial Level of Concentration $(\tau_{ij}(0))$	1.5	1.5	1.5
Maximum Number of Iteration *(T)*	200	200	200

Table 2
The experimental result for different TSP.

Instances	Algorithms	Optimal Value	Maximum Value	Minimum Value	Average Value	Variance
Patient 1	ACO	599	625.1402	605.6594	615.3998	9.7404
	IACO		637.1434	603.5242	620.3338	16.8096
	ACMPACO		**618.6746**	**602.0088**	**610.3417**	**8.3329**
Patient 2	ACO	324	356.3756	332.2638	344.3197	12.0559
	IACO		351.7483	332.2638	342.00605	**9.74225**
	ACMPACO		**338.8702**	**318.7761**	**328.82315**	10.04705
Patient 3	ACO	6532	6646.3	6552.5	6599.4	46.9
	IACO		6570.4	6478	6524.2	46.2
	ACMPACO		**6502.6**	**6437.5**	**6470.05**	**32.55**
Patient 4	ACO	529	607.4264	572.6705	590.04845	17.37795
	IACO		601.4264	570.2566	585.8415	15.5849
	ACMPACO		**575.125**	**557.125**	**566.125**	**9**
Patient 5	ACO	34303	35474	35013	35243.5	**230.5**
	IACO		35182	34580	34881	301
	ACMPACO		**35012**	**34538**	**34775**	237
Patient 6	ACO	5110	5362.1	5201.1	5281.6	80.5
	IACO		**5356.7**	5206.4	**5281.55**	**75.15**
	ACMPACO		6502.4	**5172.3**	5837.35	665.05
Patient 7	ACO	19268	26051	22031	24041	2010
	IACO		25738	21863	23800.5	1937.5
	ACMPACO		**25024**	**21266**	**23145**	**1879**
Patient 8	ACO	7806	10407	8821	9614	**793**
	IACO		10241	8242	9241.5	999.5
	ACMPACO		**9781**	**8118**	**8949.5**	831.5

values, particularly for patients 3, 2, and 1. The obtained values from ICMPACO for patients 4, 3, and 1 are 557.125, 6437.5, and 602.0088, respectively; these values are almost close to 529, 6532, and 599, respectively, which are the optimal values. It demonstrates that the ICMPACO algorithm's optimization performance is a more evident benefit. It is evident that for patients 1, 3, 4, and 7, the variance of the suggested ICMPACO method's optimization performance in solving the TSP is likewise the lowest value. This indicates that the ICMPACO algorithm is more stable than the IACO and basic ACO algorithms.

To better illustrate the optimization performance of the suggested ICMPACO technique, Fig. 4 through 11 show the optimal routes found by the ICMPACO algorithm for TSP along with their costs (route lengths). Remember that the chance of crossovers in the routes decreases as the network grows like an expanding ring. These crossings are indicative of a locally optimal gate, and the fewest patients are chosen as the optimization goals for building a multi-objective optimization model. Non-quantized processing is then used to obtain the normalized objective function as follows-

$$P = \mu_1 \left[\sum_{i=1}^{n} \sum_{l=1}^{m} S_{il}^2 + \sum_{l=1}^{m} SS_l^2 \right] + \mu_2 \sum_{i=1}^{n} \sum_{l=1}^{m} q_{il} f_l y_{il} + \mu_2 \sum_{l=1}^{m} g_i \tag{19}$$

Experimental data was found from different hospitals in Rajshahi city of Bangladesh in 2022. There were 20 operation rooms connected to the different tests required for the treatment purpose. The rooms were classified into most important (common), fairly important (common), and rarely important (common). All the patients who are not assigned to the testing room only can stay outside

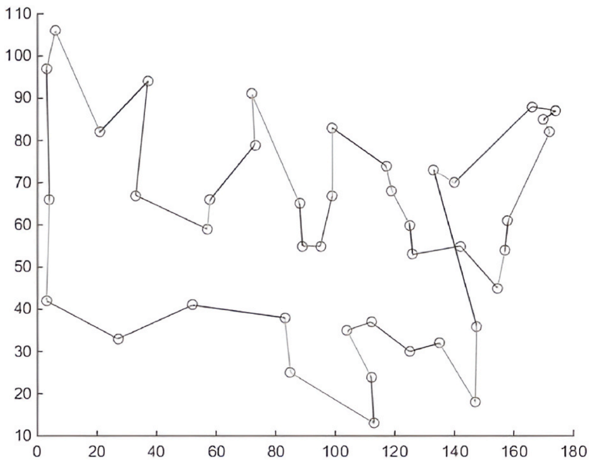

Fig. 4. Optimal route for patient 1.

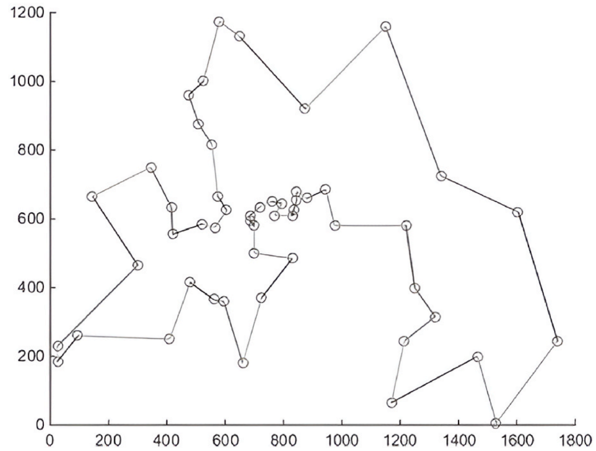

Fig. 5. Optimal route for patient 2.

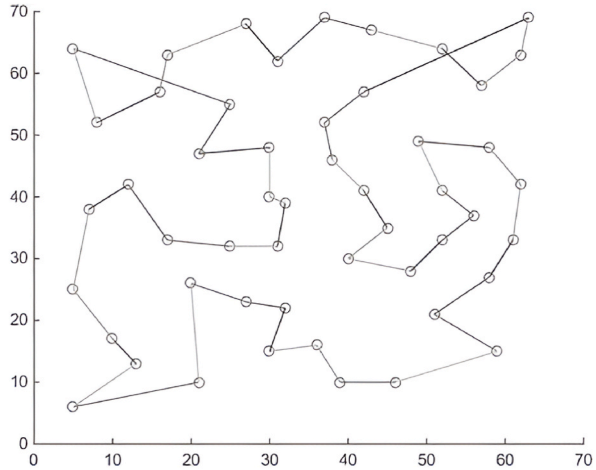

Fig. 6. Optimal route for patient 3.

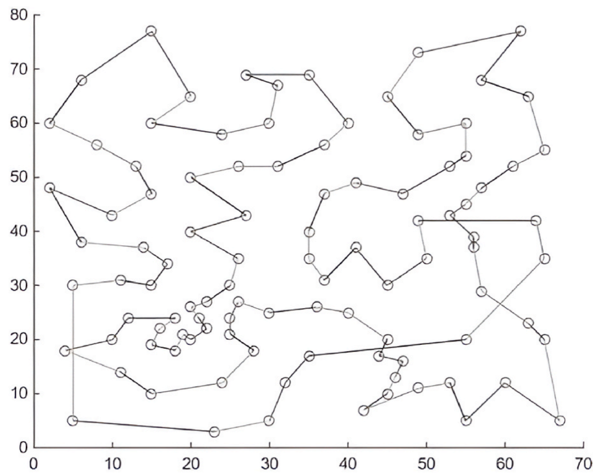

Fig. 7. Optimal route for patient 4.

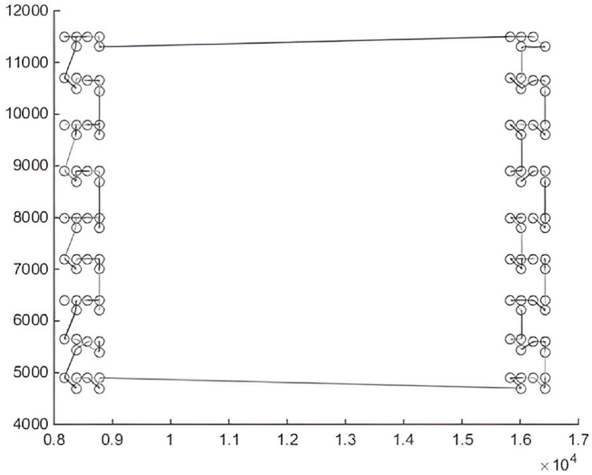

Fig. 8. Optimal route for patient 5.

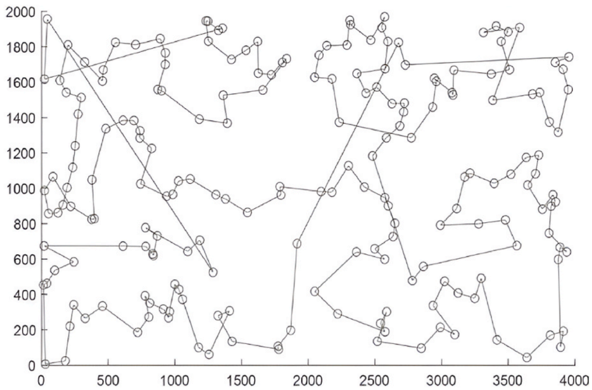

Fig. 9. Optimal route for patient 6.

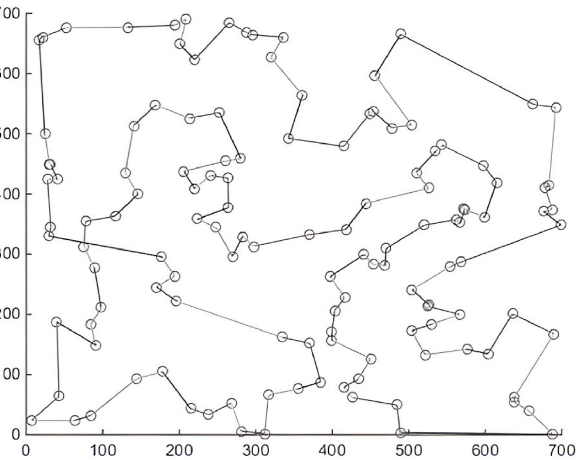

Fig. 10. Optimal route for patient 7.

the testing room. The information for the gates of the testing room is described in Table 3. The detailed information about the 20 patients is represented in Table 4. The experimental setup includes Matlab2014b, Pentium CORE i7, and 8.0 GB RAM with Windows 10.

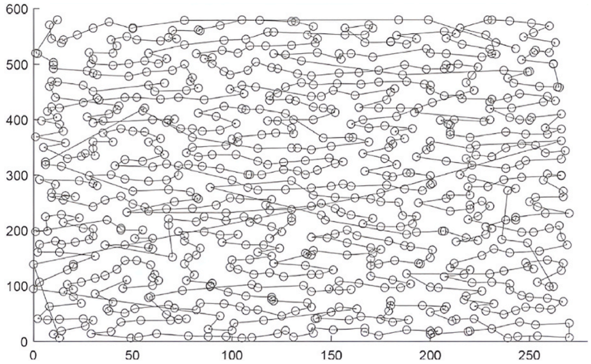

Fig. 11. Optimal route for patient 8.

Similarly, data about 157 patients were taken from several hospitals. All the patients were assigned to 20 gates of the testing room, as shown in Table 5. 83.5 % is the allotted efficiency. The number of patients is more evenly distributed for each gate based on the number of assigned results. There are sixteen gates, to which at least five patients have been allocated. Eleven patients pass via Gate 13, ten through Gate 6, nine through Gates 1, 4, 9, and 10, and seven through Gates 3, 8, and 18. Four gates require six. Each gate's patient population. There are two gates, each of which can handle five patients. Three gates park four patients.

The thorough gate assignment result demonstrates that the suggested ICMPACO method is capable of quickly and efficiently allocating these flights to 20 gates while producing the best possible assignment outcome. As a result, the suggested ICMPACO algorithm performs better optimally while tackling the gate assignment problem and possesses a more potent searching capability.

5.4. Comparison and result analysis

The gate assignment issue is solved using the similarity ACO(IACO) algorithm and the fundamental ACO method to further illustrate the optimization performance of the suggested ICMPACO algorithm. These settings are identical to those in Table 1. Five consecutive simulations were used to conduct the research. Table 6 and Fig. 10 display the outcomes of the computation and comparison.

Table 6 and Fig. 12 demonstrate how the gate assignment problem is solved using the basic ACO algorithm. The optimal assignment result is 124 patients assigned to gates, while the average assignment result is 121 patients assigned to gates with an average running time of 216.76266 s. The gate assignment problem is solved by the IACO method; the average assignment result is to allocate 123.2 patients to gates, and the average running time is 290.6041 s. The optimal assignment result is to assign 127 patients to gates. The gate assignment issue is solved by the ICMPACO method; the average assignment result is 120 patients assigned to gates, and the average running time is 322.52598 s 132 patients allocated to gates is the ideal assignment outcome. Consequently, the ICMPACO algorithm's average and best assignment results beat the results of the IACO and basic ACO algorithms when it comes to handling the gate assignment problem. This is because the three algorithms provide various outcomes. Stated otherwise, the optimal solution quality is produced by the ICMPACO algorithm. The experiment's results, however, indicate that the ICMPACO algorithm has a larger temporal complexity than the IACO and fundamental ACO algorithms.

Generally speaking, even though the ICMPACO approach takes longer to solve the gate assignment problem, the quality of its solution has enhanced when compared to the traditional ACO algorithm. The ICMPACO approach may greatly improve the efficiency of gate assignment thorough optimization issues. Consequently, the proposed ICMPACO approach is able to improve the global search performance by avoiding the local minimum value. It can effectively provide a helpful roadmap for distributing the gates.

6. Sensitivity analysis

For the determination of the impact of parameters which are important for the analysis of the algorithm, a sensitivity analysis was performed. A random instance was taken into consideration to perform this analysis. It is performed in an ACO environment with the combination of different parameters. Here, are four factors P, ϑ_{start}, ϑ_{end} *and* (d_1, d_2) were considered, as shown in Table 7. Then, this experiment was presented as an orthogonal array representation in Table 8. Each factor level has an optional value, which is presented in Table no. 9. Each of the experiments was performed independently 20 times. Standard deviation and factor values were determined and represented in the following tables. The mean is represented in Table 9. Here, the mean and standard deviation reflect the significance or impact of each parameter on others. Here, we can observe that, ϑ_{start} was considered as the most influential factor for the calculation of the ACO algorithm. From Figs. 12 and 13, it was observed that, for the factor value 10.6, the optimal value was found as the 'larger is optimal' was considered for the Taguchi function (see Fig. 14).

7. Limitations and discussion

For the analysis, data was taken from the authority of the renowned private hospital or diagnostic centre of Bangladesh. A total of

Table 3
Information about gates of testing room.

Gates No.	Types of Importance	Distance
Gate 1	Low	190
Gate 2	Medium	260
Gate 3	High	400
Gate 4	Low	333
Gate 5	Medium	384
Gate 6	High	135
Gate 7	Low	440
Gate 8	Medium	150
Gate 9	High	230
Gate 10	High	115
Gate 11	Medium	215
Gate 12	Low	535
Gate 13	High	170
Gate 14	High	585
Gate 15	Medium	450
Gate 16	High	920
Gate 17	High	1000
Gate 18	Medium	426
Gate 19	Low	265
Gate 20	High	1300

Table 4
Information about patients at hospitals.

Patient No.	Arrival Time	Departure Time	No. of Test Performed
Patient 1	0:05:00	3:12:00	05
Patient 2	0:05:00	3:15:00	03
Patient 3	0:20:00	2:03:00	03
Patient 4	0:25:00	2:27:00	03
Patient 5	0:30:00	1:25:00	02
Patient 6	0:35:00	0:50:00	01
Patient 7	0:40:00	1:07:00	01
Patient 8	0:45:00	0:55:00	01
Patient 9	0:50:00	1:52:00	02
Patient 10	0:55:00	1:05:00	01
Patient 11	1:00:00	1:55:00	01
Patient 12	1:05:00	2:04:00	02
Patient 13	1:10:00	1:45:00	01
Patient 14	1:15:00	1:33:00	01
Patient 15	1:20:00	2:42:00	02
Patient 16	1:25:00	1:45:00	01
Patient 17	1:30:00	2:22:00	02
Patient 18	1:35:00	2:12:00	01
Patient 19	1:40:00	2:31:00	01
Patient 20	1:45:00	3:23:00	02

157 patients and the authority of that hospital were given the data used as different parameters in this particular problem. Twenty gates were used for the patients here in this problem. Several gates will reduce the queue of patients. However, this was one of our limitations in the design and implementation of the model. There might be some fluctuations in the values due to the discrete data set. Additionally, more data may drive to highly accurate solution compared to this one so far. Several instances may provide more precise results for the same data set. There are some issues that arise at the time of solving the mathematical model. Some of them were avoided to minimize the design and complexity of the calculations.

8. Conclusion and future work

This work presents a revolutionary multi-population co-evolution ant colony optimization (ICMPACO) approach to handle complex, large-scale optimization problems. Its base consists of the co-evolution process, the pheromone diffusion mechanism, the pheromone update technique, and the multi-population strategy. To address the gate assignment difficulty and travelling salesman issue, the optimization efficiency of the ICMPACO algorithm is also compared with that of the IACO and basic ACO algorithms. The gate assignment problem and these TSP standard examples can be solved using the suggested ICMPACO method, which can also yield the optimum optimization value. It may quickly arrive at the optimal gate assignment result by allocating 132 patients to 20 gates of testing rooms of hospitals with an assigned efficiency of 83.5 %. As a result, compared to the ACO and IACO algorithms, the suggested ICMPACO algorithm exhibits superior optimization ability and stability.

Table 5
The result of assignment to gate.

Gate	Patients											Total
Gate 1	P15	P31	P42	P56	P68	P82	P94	P121	P156			9
Gate 2	P12	P26	P39	P100	P114	P141						6
Gate 3	P7	P27	P40	P50	P62	P93	P102					7
Gate 4	P16	P19	P24	P28	P34	P46	P60	P72	P146			9
Gate 5	P5	P77	P95	P113	P135							5
Gate 6	P3	P47	P57	P75	P80	P96	P111	P123	P138	P149		10
Gate 7	P9	P20	P38	P65	P76	P97						6
Gate 8	P17	P21	P29	P41	P132	P143	P158					7
Gate 9	P6	P30	P36	P48	P58	P78	P90	P133	P140			9
Gate 10	P4	P18	P22	P32	P43	P53	P69	P84	P131			9
Gate 11	P23	P85										2
Gate 12	P14	P51	P70	P86								4
Gate 13	P1	P33	P44	P54	P71	P87	P98	P115	P122	P134	P144	11
Gate 14	P11	P35	P45	P55	P130	P142						6
Gate 15	P2	P112	P116	P128								4
Gate 16	P10	P66	P92	P99	P124	P147						6
Gate 17	P13	P37	P49	P59	P81	P101						6
Gate 18	P8	P52	P67	P103	P117	P136	P151					7
Gate 19	P25	P118	P125	P137	P157							5
Gate 20	P61	P79	P104	P154								4
Total												132

Table 6
Comparison of results.

Serial	ACO		IACO		ICMPACO	
	Assigned Patient	Time Required to Move (s)	Assigned Patient	Time Required to Move (s)	Assigned Patient	Time Required to Move (s)
1	123	140.3432	119	145.5658	132	320.4673
2	118	120.5637	127	150.4638	131	254.7847
3	119	320.5794	123	465.6578	129	264.3647
4	121	267.4575	125	456.4584	127	445.5768
5	124	234.8695	122	234.8747	131	327.4364
Average Value	**121**	**216.76266**	**123.2**	**290.6041**	**130**	**322.52598**

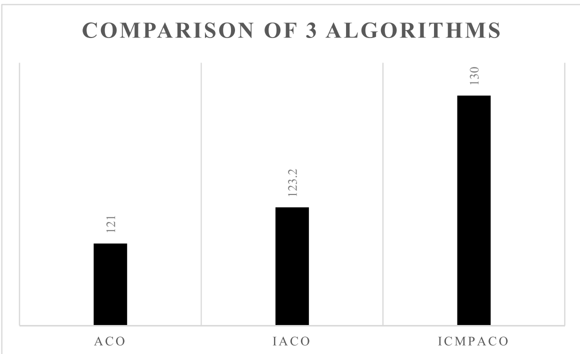

Fig. 12. Results comparison.

There are several future scopes for the research work which are presented below-

1. Because the ICMPACO technique requires more computation time to solve difficult optimization problems, more research on the algorithm is necessary to lower the time complexity.
2. The ICMPACO method will be further examined in subsequent studies and may find use in several research domains.
3. Relative comparison among the more algorithms may increase the precision of the research.
4. The algorithm may be used to show appropriateness in a particular field of research, such as inventory or transportation.

Table 7
Parameters factor level.

Parameters	Factor Level		
	1	2	3
P	$\frac{m \times n}{2}$	$m \times n$	$m \times n \times 2$
ϑ_{start}	0.1	0.2	0.3
ϑ_{end}	0.2	0.5	0.6
(d_1, d_2)	0.5	0.8	0.9

Table 8
Representation of orthogonal array.

Serial	Level of Factor				
	P	ϑ_{start}	ϑ_{end}	(d_1, d_2)	Factor Value
1	1	1	1	1	10.5
2	1	2	2	2	11.1
3	1	3	3	3	11.3
4	2	1	2	3	10.6
5	2	2	3	1	11.7
6	2	3	1	2	11.3
7	3	1	3	2	10.4
8	3	2	1	3	10.5
9	3	3	2	1	11.3

Table 9
Mean and Standard Deviation of the parameters.

Parameters	Level of Factor				Rank
	1	2	3	δ	
P	11.02	11.2	10.72	0.1812	2
ϑ_{start}	10.5	10.7	11.3	0.2276	1
ϑ_{end}	10.76	11.07	10.8	0.0726	4
(d_1, d_2)	11.16	10.87	10.76	0.1045	3

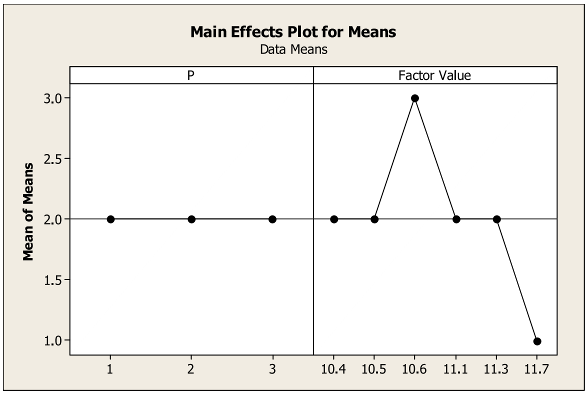

Fig. 13. Factor level trend for ACO (mean).

CRediT authorship contribution statement

Md. Limonur Rahman Lingkon: Writing – original draft, Validation, Software, Resources, Methodology, Formal analysis, Conceptualization. **Md. Sazol Ahmmed:** Writing – review & editing, Validation.

Data availability statement

The data that has been used is confidential. If required, data will be made available on request.

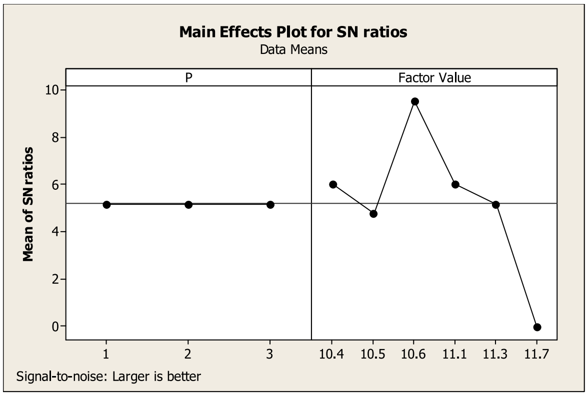

Fig. 14. Factor level trend for ACO (SN Ratio).

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors would like to acknowledge the cooperation of the management of some renowned hospitals of Rajshahi, Bangladesh, which gave us insights into our study.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.heliyon.2024.e40134.

References

[1] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 26 (1) (1996) 29–41.
[2] Liping Wang, Xin Xu, Guilin Liu, Baiyu Chen, Zhengshou Chen, A new method to estimate wave height of specified return period, Chin. J. Oceanol. Limnol. 35 (2017) 1002–1009.
[3] Q. Zhang, B. Jin, X. Wang, S. Lei, Z. Shi, J. Zhao, Q. Liu, R. Peng, The mono(Catecholamine) derivatives as iron chelators: synthesis, solution thermodynamic stability and antioxidant properties research, R. Soc. Open Sci. 5 (6) (2018), https://doi.org/10.1098/rsos.171492.
[4] H. Zhao, M. Sun, W. Deng, X. Yang, A new feature extraction method based on EEMD and multi-scale fuzzy entropy for motor bearing, Entropy 19 (1) (2017), https://doi.org/10.3390/e19010014.
[5] Zhengru Ren, Roger Skjetne, Zhen Gao, A crane overload protection controller for blade lifting operation based on model predictive control, Energies 12 (1) (2018) 50.
[6] Goran Mausa, Tihana Galinac Grbac, Co-Evolutionary multi-population genetic programming for classification in software defect prediction: an empirical case study, Appl. Soft Comput. 55 (2017).
[7] Zhizhong Liu, Dian-Hui Chu, Cheng Song, Xue Xiao, Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition, Inf. Sci. 326 (4) (2015) 315–329.
[8] W. Deng, R. Yao, H. Zhao, X. Yang, G. Li, A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm, Soft Comput. 23 (7) (2019) 2445–2462, https://doi.org/10.1007/s00500-017-2940-9.
[9] T. Li, Z. Hu, Y. Jia, J. Wu, Y. Zhou, Forecasting crude oil prices using ensemble empirical mode decomposition and sparse bayesian learning, Energies 11 (7) (2018), https://doi.org/10.3390/en11071882.
[10] Y. Zhou, T. Li, J. Shi, Z. Qian, A CEEMDAN and XGBOOST-based approach to forecast crude oil prices, Complexity (2019), https://doi.org/10.1155/2019/4392785, 2019.
[11] Miguel Cárdenas-Montes, Creating hard-to-solve instances of travelling salesman problem, Appl. Soft Comput. 71 (2018) 268–276.
[12] D. di Caprio, A. Ebrahimnejad, H. Alrezaamiri, F.J. Santos-Arteaga, A novel ant colony algorithm for solving shortest path problems with fuzzy arc weights, Alex. Eng. J. 61 (5) (2022) 3403–3415, https://doi.org/10.1016/j.aej.2021.08.058.
[13] S. Broumi, P.K. Raut, S.P. Behera, Solving shortest path problems using an ant colony algorithm with triangular neutrosophic arc weights, International Journal of Neutrosophic Science 20 (4) (2023) 128–137, https://doi.org/10.54216/IJNS.200410.
[14] C. Wu, S. Zhou, L. Xiao, Dynamic path planning based on improved ant colony algorithm in traffic congestion, IEEE Access 8 (2020) 180773–180783, https://doi.org/10.1109/ACCESS.2020.3028467.

[15] Leandro Coelho, Diego Luis de Andrade Bernert, A modified ant colony optimization algorithm based on differential evolution for chaotic synchronization, Expert Syst. Appl. 37 (6) (2010) 4198–4203.
[16] Juan Rada-Vilela, Manuel Chica, Oscar Cordon, S. Damas, A comparative study of multi-objective ant colony optimization algorithms for the time and space assembly line balancing problem, Appl. Soft Comput. 13 (11) (2013) 4370–4382.
[17] Zhengru Ren, Roger Skjetne, Zhiyu Jiang, Zhen Gao, Integrated GNSS/IMU hub motion estimator for offshore wind turbine blade installation, Mech. Syst. Signal Process. 123 (4) (2019) 222–243.
[18] S. Pal, A. Roy, P. Shivakumara, U. Pal, Adapting a swin transformer for license plate number and text detection in drone images, Artificial Intelligence and Applications 1 (3) (2023) 145–154, https://doi.org/10.47852/bonviewaia3202549.
[19] R.G.M. Helali, An exploratory study of factors affecting research productivity in higher educational institutes using regression and deep learning techniques, Artificial Intelligence and Applications (2023), https://doi.org/10.47852/bonviewaia3202660.
[20] T. Akande, O.A. Alabi, S.A. Ajagbe, A deep learning-based CAE approach for simulating 3D vehicle wheels under real-world conditions, Artificial Intelligence and Applications (2024), https://doi.org/10.47852/bonviewaia42021882.
[21] P.B. Myszkowski, M.E. Skowroński, Ł.P. Olech, K. Oślizło, Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem, Soft Comput. 19 (12) (2015) 3599–3619, https://doi.org/10.1007/s00500-014-1455-x.
[22] W. Deng, H. Zhao, X. Yang, D. Li, Y. Li, J. Liu, Research on a robust multi-objective optimization model of gate assignment for hub airport, Transportation Letters 10 (4) (2018) 229–241, https://doi.org/10.1080/19427867.2016.1252876.
[23] Kamil Krynicki, Michael E. Houle, Javier Jaen, An efficient ant colony optimization strategy for the resolution of multi-class queries, Knowl. Base Syst. 105 (2016) 96–106.
[24] H. Fu, Z. Li, Z. Liu, Z. Wang, Research on big data digging of hot topics about recycled water use on micro-blog based on particle swarm optimization, Sustainability 10 (7) (2018), https://doi.org/10.3390/su10072488.
[25] Mousa Marzband, Ebrahim Yousefnejad, Andreas Sumper, José Luis Domínguez-García, Real time experimental implementation of optimum energy management system in standalone Microgrid by using multi-layer ant colony optimization, Int. J. Electr. Power Energy Syst. 75 (2016) 265–274.
[26] A. Khan, A.R. Baig, Multi-objective feature subset selection using mRMR based enhanced ant colony optimization algorithm (mRMR-EACO), J. Exp. Theor. Artif. Intell. 28 (6) (2016) 1061–1073.
[27] W. Deng, S. Zhang, H. Zhao, X. Yang, A novel fault diagnosis method based on integrating empirical wavelet transform and fuzzy entropy for motor bearing, IEEE Access 6 (2018) 35042–35056.
[28] F. Huang, C. Yao, W. Liu, Y. Li, X. Liu, Landslide susceptibility assessment in the Nantian area of China: a comparison of frequency ratio model and support vector machine, Geomatics, Nat. Hazards Risk 9 (1) (2018) 919–938.
[29] Le Kang, Hui Ling Du, Hao Zhang, Wan Li Ma, Systematic research on the application of steel slag Resources under the background of big data, Complexity 2018 (2018).
[30] Liming Shen, Huiling Chen, Zhe Yu, Wenchang Kang, Bingyu Zhang, Huaizhong Li, Bo Yang, Bo Yang, Evolving support vector machines using fruit fly optimization for medical data classification, Knowl. Base Syst. 96 (2016) 61–75.
[31] P. Vijayalakshmi, Sharmila Anand John Francis, J. Abraham Dinakaran, A robust energy efficient ant colony optimization routing algorithm for multi-hop ad hoc networks in MANETs, Wireless Network 22 (2015) 2081–2100.
[32] Pawan K. Tiwari, Deo Vidyarthi, Improved auto control ant colony optimization using lazy ant approach for grid scheduling problem, Future Generat. Comput. Syst. 60 (2016) 78–89.
[33] Hossein Sharifipour, Mojtaba Shakeri, H. Haghighi, Structural test data generation using a memetic ant colony optimization based on evolution strategies, Swarm Evol. Comput. 40 (2017) 76–91.
[34] Khaled Akka, Farid Khaber, Mobile robot path planning using an improved ant colony optimization, Int. J. Adv. Rob. Syst. 15 (3) (2018).
[35] Q. Zhang, H. Chen, J. Luo, Y. Xu, C. Wu, C. Li, Chaos enhanced bacterial foraging optimization for global optimization, IEEE Access 6 (2018) 64905–64919.
[36] J. Pei, X. Liu, W. Fan, P.M. Pardalos, S. Lu, A hybrid BA-VNS algorithm for coordinated serial-batching scheduling with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers, Omega (United Kingdom) 82 (2019) 55–69, https://doi.org/10.1016/j.omega.2017.12.003.
[37] E. Hassan, T. Abd El-Hafeez, M.Y. Shams, Optimizing classification of diseases through language model analysis of symptoms, Sci. Rep. 14 (1) (2024), https://doi.org/10.1038/s41598-024-51615-5.
[38] D.A. Abdel Hady, T. Abd El-Hafeez, Predicting female pelvic tilt and lumbar angle using machine learning in case of urinary incontinence and sexual dysfunction, Sci. Rep. 13 (1) (2023), https://doi.org/10.1038/s41598-023-44964-0.
[39] E.H.I. Eliwa, A.M. El Koshiry, T. Abd El-Hafeez, H.M. Farghaly, Utilizing convolutional neural networks to classify monkeypox skin lesions, Sci. Rep. 13 (1) (2023), https://doi.org/10.1038/s41598-023-41545-z.
[40] H. Mamdouh Farghaly, M.Y. Shams, T. Abd El-Hafeez, Hepatitis C Virus prediction based on machine learning framework: a real-world case study in Egypt, Knowl. Inf. Syst. 65 (6) (2023) 2595–2617, https://doi.org/10.1007/s10115-023-01851-4.
[41] T. Abd El-Hafeez, M.Y. Shams, Y.A.M.M. Elsevier, H.M. Farghaly, A.E. Hassanien, Harnessing machine learning to find synergistic combinations for FDA-approved cancer drugs, Sci. Rep. 14 (1) (2024), https://doi.org/10.1038/s41598-024-52814-w.
[42] A. Omar, T. Abd El-Hafeez, Optimizing epileptic seizure recognition performance with feature scaling and dropout layers, Neural Comput. Appl. 36 (6) (2024) 2835–2852, https://doi.org/10.1007/s00521-023-09204-6.
[43] W. Deng, R. Chen, B. He, Y. Liu, L. Yin, J. Guo, A novel two-stage hybrid swarm intelligence optimization algorithm and application, Soft Comput. 16 (10) (2012) 1707–1722, https://doi.org/10.1007/s00500-012-0855-z.
[44] Jie luo, Huiling Chen, Qian Zhang, Yueting Xu, An improved grasshopper optimization algorithm with application to financial stress prediction, Appl. Math. Model. 64 (2018) 654–668.
[45] Mukund Nilakantan Janardhanan, Zixiang Li, Qiuhua Tang, Peter Nielsen, Multi-objective co-operative co-evolutionary algorithm for minimizing carbon footprint and maximizing line efficiency in robotic assembly line systems, J. Clean. Prod. 156 (2017) 124–136.
[46] F. Zhang, T. Lei, J. Li, X. Cai, X. Shao, J. Chang, F. Tian, Real-time Calibration and Registration Method for Indoor Scene with Joint Depth and Color Camera 32 07 (2018) 1854021, https://doi.org/10.1142/S0218001418540204.
[47] H. Zhao, R. Yao, L. Xu, Y. Yuan, G. Li, W. Deng, Study on a novel fault damage degree identification method using high-order differential mathematical morphology gradient spectrum entropy, Entropy 20 (9) (2018), https://doi.org/10.3390/e20090682.
[48] F.Q. Zhang, Z.W. Wang, J. Chang, J. Zhang, F. Tian, A fast framework construction and visualization method for particle-based fluid, EUPASIP J. Image Video Process 17 (1) (2017) 79.
[49] F.R. Sun, Y.D. Yao, X.F. Li, The heat and mass transfer characteristics of superheated steam in horizontal wells with toe-point injection technique, Energy 8 (2018) 1295–1302.
[50] F. Zhang, T. Lei, J. Li, X. Cai, X. Shao, J. Chang, F. Tian, Real-time calibration and registration method for indoor scene with joint depth and color camera, Int. J. Pattern Recogn. Artif. Intell. 32 (7) (2018) 1–26.
[51] A.E. Yildirim, A. Karci, Applications of artificial atom algorithm to small-scale traveling salesman problems, Soft Comput. 22 (22) (2018) 7619–7631.
[52] D.Q. Wu, J.Z. Huo, G.F. Zhang, W.H. Zhang, Minimization of logistics cost and carbon emissions based on quantum particle swarm optimization, Sustainability 9 (1) (2018) 3791.
[53] S. Guo, R. Chen, M. Wei, H. Li, Y. Liu, Ensemble data reduction techniques and Multi-RSMOTE via fuzzy integral for bug report classification, IEEE Access 6 (2018) 45934–45950, https://doi.org/10.1109/ACCESS.2018.2865780.
[54] W. Deng, H. Zhao, L. Zou, G. Li, X. Yang, D. Wu, A novel collaborative optimization algorithm in solving complex optimization problems, Soft Comput. 21 (15) (2017) 4387–4398, https://doi.org/10.1007/s00500-016-2071-8.

[55] Shikai Guo, Rong Chen, Hui Li, Jian Gao, Crowdsourced web application testing under real-time constraints, Int. J. Software Eng. Knowl. Eng. 28 (6) (2018) 751–779.
[56] I.D.I.D. Ariyasingha, T.G.I. Fernando, Performance analysis of the multi-objective ant colony optimization algorithms for the traveling salesman problem, Swarm Evol. Comput. 23 (2015) 11–26.
[57] panelQuan-Ke Pan, An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling, Eur. J. Oper. Res. 250 (3) (2016) 702–714.