JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY
https://e-journal.uum.edu.my/index.php/jict

How to cite this article:
Muthana, S. A., & Ku-Mahamud, K. R. (2023). Taguchi-Gray relational analysis method for parameter tuning of multi-objective pareto ant colony system algorithm. *Journal of Information and Communication Technology, 22*(2), 149-181. https://doi.org/10.32890/jict2023.22.2.1

Taguchi-Gray Relational Analysis Method for Parameter Tuning of Multi-objective Pareto Ant Colony System Algorithm

*[1]Shatha Abdulhadi Muthana, [2]Ku Ruhana Ku-Mahamud
[1]General Company of Electricity Production South Region, Ministry of Electricity, Iraq
[2]School of Computing, Universiti Utara Malaysia, Malaysia
[2]Shibaura Institute of Technology, Tokyo, Japan

*shatha_muthana@yahoo.com
ruhana@uum.edu.my
*Corresponding author

Received: 30/6/2022 Revised: 3/11/2022 Accepted: 29/11/2022 Published: 3/4/2023

ABSTRACT

In any metaheuristic, the parameter values strongly affect the efficiency of an algorithm's search. This research aims to find the optimal parameter values for the Pareto Ant Colony System (PACS) algorithm, which is used to obtain solutions for the generator maintenance scheduling problem. For optimal maintenance scheduling with low cost, high reliability, and low violation, the parameter values of the PACS algorithm were tuned using the Taguchi and Gray Relational Analysis (Taguchi-GRA) method through search-based approach.

The new parameter values were tested on two systems. i.e., 26- and 36-unit systems for window with operational hours [3000-5000]. The gray relational grade (GRG) performance metric and the Friedman test were used to evaluate the algorithm's performance. The Taguchi-GRA method that produced the new values for the algorithm's parameters was shown to be able to provide a better multi-objective generator maintenance scheduling (GMS) solution. These values can be benchmarked in solving multi-objective GMS problems using the multi-objective PACS algorithm and its variants.

Keywords: Optimization, Scheduling, Taguchi method, Gray Relational Analysis, Generator maintenance.

INTRODUCTION

Parameter tuning enhances an algorithm's flexibility and robustness because parameters strongly affect efficient and effective search for solutions (Negulescu, 2017; Talbi, 2009). However, careful initialization is required (Sagban, 2016; Talbi, 2009). Optimal values for the parameters mainly depend on the problem and the instance it deals with, which also depends on the search time needed to solve the problem (Talbi, 2009). Two different strategies exist for parameter tuning, comprising offline and online parameter tunings (Negulescu, 2017; Sagban, 2016; Talbi, 2009).

Under offline parameter tuning, different parameter values are fixed ahead of the metaheuristic execution (Talbi, 2009). Traditionally, "trial and error" is applied mostly in the execution of offline tuning. In addition, under the offline parameter tuning strategy, the Taguchi method is widely used in engineering analysis. The greater advantages of this method are saving efforts in conducting experiments, saving experimental time, and discovering significant factors quickly (Manikandan et al., 2015). Recently, increasing effort has been made to allow the tuning of algorithm parameters to be through a search-based approach (Negulescu, 2017). Under online tuning, in the process of metaheuristic execution, the control of parameters involves an updated dynamic or adaptive approach (Talbi, 2009). Under the dynamic update approach, parameter value changes are executed without consideration of the search progress, and parameter value update is carried out in a random or deterministic order. However, in the adaptive approach, values are changed in accordance with the search progress, which is mostly through memory of the search (Talbi, 2009).

Conventionally, "trial and error" is applied primarily in the execution of offline tuning. Nevertheless, this method is effortful and time-consuming without guaranteeing optimal values (Negulescu, 2017). The tuning parameters with the Taguchi method, which is an offline method, has shown to produce a robust design with less experimentation (Kolahan & Azadi Moghaddam, 2015). As for online parameter tuning methods, these methods generally use feedback from the optimization process to continuously update their parameters. For instance, when increasing the size of the dataset, the algorithm may not be able to improve its solutions.

In this research, the Taguchi and Gray Relational Analysis (Taguchi-GRA) method is used for tuning parameter values in a Pareto Ant Colony System (PACS) algorithm (Muthana & Ku-Mahamud, 2021; 2022). This algorithm is used to solve the multi-objective generator maintenance scheduling (GMS) problem. A review of previous tuning methods on variants of Ant Colony Optimization (ACO) algorithms is presented in the second section, followed by the description of the Taguchi-GRA method. The experimental results and discussion are then presented, followed by the conclusion and future research in the last section.

RELATED LITERATURE

The ACO algorithms' behaviors largely depend on the values associated with the parameters (López-Ibáñez et al., 2015; Negulescu, 2017; Yasear & Ku-Mahamud, 2021). Several studies have been carried out for developing parameterization strategies to ensure that the metaheuristics trade-off is achieved between exploration and exploitation. The aim is to discover global optimal solutions in the quickest time possible (Zheng et al., 2017). Two different strategies exist for parameter tuning, namely offline and online parameter tunings (Negulescu, 2017; Sagban, 2016).

Under offline parameter tuning, different parameter values are fixed ahead of the metaheuristic execution (Talbi, 2009). The "trial and error" method is applied mostly in the execution of offline tuning. This process is considered to be human-intensive, time-consuming, and error-prone and, in most instances, results in the uneven tuning of different algorithms (Negulescu, 2017). Negulescu (2017) used the normalization method to regulate the parameter values of the Elitist Ant System algorithm. The advantage of this method is that it spares computational time for otherwise running empirical test runs for determining a good set of parameter values, and the determination of parameter values can be extrapolated for other similar maps.

The Taguchi method, which is used for parameter design, is based on the theory that the experimental designs use orthogonal matrices, making it possible to easily determine the effects of the variables (Sihem & Benmansour, 2018). The Taguchi method is an important tool for robust design in which the best setting of the control factors (parameters) is determined. The two major tools used in the Taguchi methodology are: (1) orthogonal array and (2) signal-to-noise ratio (S/N) analysis (Vinay & Sridharan, 2013). Orthogonal arrays are used to analyze design parameters, and the S/N ratio measures production quality (Yuan-Kang et al., 2013). However, the Taguchi method does not consider the search-based approach. Instead, it uses feedback from the optimization process by analyzing the results of the objective functions, which allows it to obtain optimal or near optimal parameter values even in big system sizes.

Under online tuning, in the process of metaheuristic execution, the control of parameters involves updated dynamic or adaptive approaches (Talbi, 2009). Under the dynamic update approach, parameter value changes are executed without consideration of the search progress, and a random or deterministic update of the parameter values is performed. Differently, in the adaptive approach, the values are changed according to the search progress, mostly through memory of the search. The most generic strategies of the online approach are pre-scheduled strategy, adaptive strategy, and self-adaptive strategy, which include pure self-adaptive strategy and search-adaptive strategy. The pre-scheduled strategy involves observation of the problem from an offline perspective. The substitution of static parameters is achieved through either deterministic or randomized functions. The functions depend on the volumes of algorithm iterations, or computational in which the change of parameter values is achieved in the optimization process, based on some programmed rules, while any feedback is ignored during the search (Sagban, 2016). An adaptive strategy approach uses feedback from the optimization process for the continuous update of parameters (Drozdik et al., 2015). In this strategy, ACO algorithm changes are made to parameters based on specific rules that consider the importance of ACO algorithm search behavior (Sagban, 2016).

With the self-adaptive strategy, a further possibility is to have parameter modification by the algorithm itself during the run time, this approach is called self-adaptation (Sagban, 2016). Many strategies for adaptive ACO are classified within the components of self-adaptive strategies. The algorithm utilizes itself instead of using other search methods for adapting its parameters. This strategy is classified as the "pure self-adaptive strategy", which is a way of implicitly adapting the ACO parameters where the algorithm utilizes itself for adapting its parameters (Sagban, 2016). Meanwhile, the "search-adaptive strategy" implicitly adapts the ACO algorithm's parameters in which the algorithm utilizes alternative search methods to adapt its parameters (Sagban, 2016).

Table 1 presents the summary of several studies for different application domains that include the offline and online tuning strategies in ACO algorithm variants from 2017 to 2022. The variants of ACO include the Ant System (AS), Elitist Ant System (EAS), Rank-based Ant System (ASrank), Ant Colony System (ACS), and Max-Min Ant System (MMAS). This research focuses on parameter tuning for the Pareto Ant Colony System (PACS) algorithm, which is a variant of the ACS algorithm. The parameters that are used in ACO variants are: i) (α) and (β) to control the relative importance of pheromone trails and heuristic information on the decision probabilities, respectively; ii) uniform distributed variable (q); iii) evaporation parameters (ρ and ξ); iv) number of iterations (S); v) parameter (m) representing number of ants and neighbors, respectively; vi) (δ) effectiveness factor of pheromone deviation from the upper bound of pheromone trail; vii) ($\Delta\tau xy$) represents deposited pheromone amount; viii) ($P0$) represents the threshold probability, which is selected out of the calculated probability values; ix) (*ptries*) is the number of attempts of the randomized packing heuristic; x) (*localsearch*) controls whether and what local search procedure to apply; and finally, xi) (P) is the transfer probability parameter.

Table 1

Parameter Tuning Strategies in ACO

Authors	Type of Strategy	ACO Variant	Parameter	Type of Problem
Sihem and Benmansour (2018)	Offline strategy/ Taguchi	AS	*α, β, ρ, q*	Electric power system reliability
Tirkolaee al. (2019)	Offline strategy/ Taguchi	MMAS	*α, β, ρ, m, S, δ*	Multi-trip capacitated arc routing
Ragmani al. (2019)	Offline strategy/ Taguchi	AS	*α, β, ρ, m, S*	Identification of the optimal configuration of virtual machine placement
Lyu et al. (2020)	Offline strategy/ Taguchi	ACS	*P, ρ, m, S*	Tilt quad rotor problem
Ragmani et al. (2020)	Offline strategy/ Taguchi	ACS	*α, β, ρ, m, S*	Scheduling/ Virtual machine
Lezama et al. (2020)	Offline strategy/ Trial and error	ACS	*α, β, ρ, m, S*	Local electricity markets
Ankita and Sahana (2019)	Online strategy/ Pre-scheduled	ACS	*Δτxy, P0*	Scheduling/ Grid environment
Mavrovouniotis et al. (2017)	Online strategy/ Pre-scheduled	MMAS	*m*	Dynamic traveling salesman problem
Zheng et al. (2017)	Online strategy/ Adaptive	ASrank	*α*	Water distribution system design problems
Chagas and Wagner (2020)	Online strategy/ Pure self-adaptive	AS	*m, α, β, ρ, ptries*	Thief orienteering problem

(continued)

Authors	Type of Strategy	ACO Variant	Parameter	Type of Problem
Chagas and Wagner (2022)	Online strategy/ Search-adaptive	MMAS	*m, α, β, ρ, localsearch, ptries*	Thief orienteering problem
Wang and Han (2021)	*Online strategy/ Search-adaptive*	*ACS*	*α, β*	*Traveling salesman problem*
Han et al. (2021)	*Online strategy/ Search-adaptive*	*ACS*	*α, β, ρ, ξ*	*Assembly sequence planning*
Ariyaratne and Fernando (2018)	*Online strategy/ Search-adaptive*	*ACS*	*Specific parameters*	*Traveling salesman problem*

In summary, most of the studies used online parameter tuning strategies because this type of method has shown its efficiency in obtaining better results with less computational time although it is not always the best for some problems. Most of the tuned parameters are the standard parameters for ACO and its variants, while the parameters that are not usually tuned are *δ*, *localsearch, ptries, P0*, and *P*, which are used in the ACO algorithms for a special purpose. The Taguchi method proved its proficiency to calibrate parameters for ACS by optimizing solutions for single objective GMS (Fattahi et al., 2014). In this research, the Taguchi-GRA method is adopted to enhance the work of Muthana and Ku-Mahamud (2021; 2022) in finding the optimal parameter values for the proposed PACS algorithm to optimize solutions for multi-objective GMS.

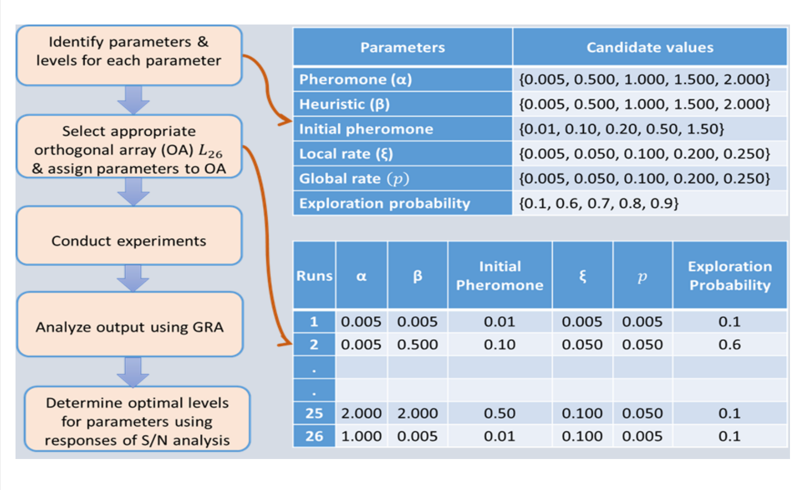
TAGUCHI-GRAY RELATIONAL ANALYSIS METHOD FOR DETERMINING OPTIMAL PARAMETERS

The PACS algorithm proposed by Muthana and Ku-Mahamud (2021; 2022) was used to obtain solutions for the multi-objective GMS problem in electrical power systems. In this section, the Taguchi-GRA method was employed to determine the optimal value for each parameter in PACS for optimal maintenance scheduling. The Taguchi method was used to configure the design of parameter values for the PACS algorithm, while GRA analyzed the output from Taguchi to obtain the gray relational grade (GRG) values. GRG converted the multi-objective PACS solution to a single solution to be evaluated by the signal-to-noise ratio (S/N) approach.

There were five main steps for implementing the Taguchi-GRA method as shown in Figure 1. The first step was to identify the factors (parameters) and levels (values) for each parameter. In this research, six parameters with five levels of values for each parameter were tested. The second step was to select an appropriate orthogonal array and assign the parameters to the array. The third step was to assign the parameters to the columns of the orthogonal array and conduct the experiments. The fourth step was to analyze the output from the experiments in the previous step using the GRA method. The fifth step determined the best levels for parameters using responses of the S/N analysis that determined the best levels for parameters.

Figure 1

Taguchi-GRA Method



The initial values (or candidate values) that were used in the experiments for testing the PACS parameters are displayed in Table 2. These parameters and values were constantly being used when scheduling the maintenance of generating units in electrical power systems using the ACS algorithm and multi-objective scheduling using the PACS algorithm (Berrichi et al., 2010; Fattahi et al., 2014).

Table 2

Test Parameter Values

Parameters	Candidate Values
Pheromone power (α)	{0.005, 0.500, 1.000, 1.500, 2.000}
Heuristic power (β)	{0.005, 0.500, 1.000, 1.500, 2.000}
Initial pheromone (τ*0*)	{0.01, 0.10, 0.20, 0.50, 1.50}
Local rate (ξ)	{0.005, 0.050, 0.100, 0.200, 0.250}
Global rate (ρ)	{0.005, 0.050, 0.100, 0.200, 0.250}
Exploration probability	{0.1, 0.6, 0.7, 0.8, 0.9}

Experimental Design for Taguchi

Tables 3 and 4 show the results of using the Taguchi design on the initial values of the PACS algorithm for the 26-unit and 36-unit systems, respectively. These results were obtained from ten runs. The [3,000–5,000] maintenance window were used because the results for this window had been previously obtained in all the unit systems. There were 25 rows in the tables as the results of the Taguchi method that used six parameters in the proposed PACS and five levels of values for each parameter as proposed by Berrichi et al. (2010) in their research on multi-objective scheduling. The experimental design for six controllable parameters with five levels was organized by Taguchi in an orthogonal array of 25 rows (i.e., $L_{25}(5^6)$). However, in this research, an additional combination was added, giving the final orthogonal array of 26 rows (L_{26}). The additional combination was to provide an extra alternative solution (Fattahi et al., 2014).

Table 3

Experimental Layout and Performance Results for 26 Units

Runs	Alpha	Beta	Initial Pheromone	Global Rate	Local Rate	Exploration Probability	Cost	Reliability	Violation
1	0.005	0.005	0.01	0.005	0.005	0.1	177,734,787.36	2,004,062.00	0.00
2	0.005	0.500	0.10	0.050	0.050	0.6	175,612,105.31	1,990,910.00	0.00
3	0.005	1.000	0.20	0.100	0.100	0.7	175,874,313.74	2,005,294.00	0.00
4	0.005	1.500	0.50	0.200	0.200	0.8	174,977,256.39	1,957,449.00	0.00
5	0.005	2.000	1.50	0.250	0.250	0.9	174,993,692.11	1,940,853.00	0.00
6	0.500	0.005	0.10	0.100	0.200	0.9	176,973,123.32	2,007,187.00	0.00
7	0.500	0.500	0.20	0.200	0.250	0.1	177,070,927.64	2,002,925.00	0.00
8	0.500	1.000	0.50	0.250	0.005	0.6	174,329,215.10	1,965,233.00	0.00
9	0.500	1.500	1.50	0.005	0.050	0.7	176,383,970.09	2,008,316.00	0.00
10	0.500	2.000	0.01	0.050	0.100	0.8	174,748,668.44	1,936,622.00	0.00
11	1.000	0.005	0.20	0.250	0.050	0.8	177,492,428.34	2,013,578.00	0.00
12	1.000	0.500	0.50	0.005	0.100	0.9	177,404,500.65	2,012,457.00	0.00
13	1.000	1.000	1.50	0.050	0.200	0.1	176,639,853.11	1,998,769.00	0.00
14	1.000	1.500	0.01	0.100	0.250	0.6	176,348,399.40	2,003,844.00	0.00
15	1.000	2.000	0.10	0.200	0.005	0.7	176,604,890.35	2,001,763.00	0.00
16	1.500	0.005	0.50	0.050	0.250	0.7	175,898,339.19	1,994,786.00	0.00
17	1.500	0.500	1.50	0.100	0.005	0.8	177,686,007.09	2,016,489.00	0.00

(continued)

Runs	Alpha	Beta	Initial Pheromone	Global Rate	Local Rate	Exploration Probability	Cost	Reliability	Violation
18	1.500	1.000	0.01	0.200	0.050	0.9	172,500,403.87	1,920,825.00	0.00
19	1.500	1.500	0.10	0.250	0.100	0.1	176,562,189.37	1,993,947.00	0.00
20	1.500	2.000	0.20	0.005	0.200	0.6	178,145,200.37	2,003,357.00	0.00
21	2.000	0.005	1.50	0.200	0.100	0.6	175,799,834.32	2,001,297.00	0.00
22	2.000	0.500	0.01	0.250	0.200	0.7	172,716,126.54	1,947,006.00	0.00
23	2.000	1.000	0.10	0.005	0.250	0.8	176,777,038.17	2,016,810.00	0.00
24	2.000	1.500	0.20	0.050	0.005	0.9	175,606,860.65	2,004,525.00	0.00
25	2.000	2.000	0.50	0.100	0.050	0.1	176,678,994.75	1,997,868.00	0.00
26	1.000	0.005	0.01	0.100	0.005	0.1	175,125,928.53	2,013,313.00	0.00

Table 4

Experimental Layout and Performance Results for 36 Units

Runs	Alpha	Beta	Initial pheromone	Global Rate	Local Rate	Exploration Probability	Cost	Reliability	Violation
1	0.005	0.005	0.01	0.005	0.005	0.1	337,530,904.79	2,430,417.00	0.00
2	0.005	0.500	0.10	0.050	0.050	0.6	344,370,076.96	2,411,039.00	0.00
3	0.005	1.000	0.20	0.100	0.100	0.7	339,708,432.52	2,422,805.00	0.00
4	0.005	1.500	0.50	0.200	0.200	0.8	338,411,921.54	2,413,950.00	0.00
5	0.005	2.000	1.50	0.250	0.250	0.9	340,154,972.04	2,430,895.00	0.00
6	0.500	0.005	0.10	0.100	0.200	0.9	343,568,714.64	2,397,513.00	0.00
7	0.500	0.500	0.20	0.200	0.250	0.1	337,241,714.00	2,420,037.00	0.00
8	0.500	1.000	0.50	0.250	0.005	0.6	338,057,763.28	2,431,945.00	0.00
9	0.500	1.500	1.50	0.005	0.050	0.7	338,593,559.20	2,419,424.00	0.00
10	0.500	2.000	0.01	0.050	0.100	0.8	342,903,128.20	2,414,014.00	0.00
11	1.000	0.005	0.20	0.250	0.050	0.8	340,164,743.28	2,422,999.00	0.00
12	1.000	0.500	0.50	0.005	0.100	0.9	344,398,665.90	2,423,879.00	0.00
13	1.000	1.000	1.50	0.050	0.200	0.1	343,496,705.20	2,408,725.00	0.00
14	1.000	1.500	0.01	0.100	0.250	0.6	336,739,805.60	2,426,472.00	0.00
15	1.000	2.000	0.10	0.200	0.005	0.7	335,755,180.10	2,420,809.00	0.00
16	1.500	0.005	0.50	0.050	0.250	0.7	338,158,071.40	2,414,480.00	0.00
17	1.500	0.500	1.50	0.100	0.005	0.8	337,827,703.30	2,404,611.00	0.00

(continued)

Runs	Alpha	Beta	Initial pheromone	Global Rate	Local Rate	Exploration Probability	Cost	Reliability	Violation
18	1.500	1.000	0.01	0.200	0.050	0.9	337,217,613.60	2,424,854.00	0.00
19	1.500	1.500	0.10	0.250	0.100	0.1	336,614,327.00	2,423,084.00	0.00
20	1.500	2.000	0.20	0.005	0.200	0.6	345,606,538.00	2,415,638.00	0.00
21	2.000	0.005	1.50	0.200	0.100	0.6	340,003,610.60	2,421,594.00	0.00
22	2.000	0.500	0.01	0.250	0.200	0.7	339,179,004.10	2,421,193.00	0.00
23	2.000	1.000	0.10	0.005	0.250	0.8	342,759,279.50	2,404,671.00	0.00
24	2.000	1.500	0.20	0.050	0.005	0.9	335,595,333.10	2,412,826.00	0.00
25	2.000	2.000	0.50	0.100	0.050	0.1	340,414,956.50	2,407,707.00	0.00
26	1.000	0.005	0.01	0.100	0.005	0.1	336,980,517.06	2,414,183.00	0.00

The results of cost, reliability, and violation were converted to a single value (i.e., GRG) using the GRA method. This process involved data normalization, deviation sequences, and gray relational coefficient (GRC). The obtained GRG values were used in the S/N analysis to produce the new parameter values for the PACS algorithm. The following sections describe the steps in implementing GRA, followed by the S/N analysis.

Gray Relational Analysis Method and Signal-to-Noise Ratio Analysis for Optimal Parameters Values

In this section, the use of the GRA method and S/N analysis in determining the optimal parameters to be used in the PACS algorithm for GMS are discussed. The gray relational grade (GRG) of the objective functions of the PACS solution algorithm was considered as the response variable of the design. In contrary to Fattahi et al. (2014), the operation cost objective function of the solution algorithm was considered as the response variable of the design. The design of ACS parameter values proposed by Fattahi et al. (2014) considered the S/N analysis to determine the best set of parameter levels based on a single objective only. Here, the problem under consideration had three distinctive conflicting objectives. In many practical cases, it is desirable to make a balance among these objectives. To overcome this shortcoming of S/N, this research employed GRA to turn all three objectives into a single criterion called GRGs (Kolahan & Azadi Moghaddam, 2015). The following subsections represent the steps in implementing GRA and S/N ratio analysis.

Data Normalization

Data normalization was performed on the values obtained from the Taguchi design (i.e., Tables 3 and 4). Numerical data were normalized between zero and one. In this research, the normalized values (x_{ij}) for cost and convenience (violation) objective functions were calculated based on Equation 1 (Jozić et al., 2015):

$$\boldsymbol{x_{ij}} = \frac{\mathbf{max}(\boldsymbol{y_{ij}}) - \boldsymbol{y_{ij}}}{\boldsymbol{max}(\boldsymbol{y_{ij}}) - \boldsymbol{min}(\boldsymbol{y_{ij}})} \tag{1}$$

where y_{ij} is the value for cost and convenience, and *max* (*yij*) and *min* (y_{ij}) are the maximum and minimum values for cost and convenience, respectively. This equation was used because a smaller value of cost or violation indicated a better result for the objective function. However, in the case of reliability, a larger value specified a better result for the objective function. Therefore, the normalization calculation reliability is based on Equation 2 (Jozić et al., 2015):

$$x_{ij} = \frac{y_{ij} - \min (y_{ij})}{max(y_{ij}) - min(y_{ij})} \tag{2}$$

where y_{ij} is the value for reliability, and *max* (*yij*) and *min* (y_{ij}) are the maximum and minimum values for reliability. The normalized values of cost, reliability, and violation are demonstrated in Tables 5 and 6.

Table 5

Normalization Experimental Results for 26 Units

Runs	Cost	Reliability	Violation
1	0.072706431	0.867187581	-
2	0.448748694	0.730166172	-
3	0.402297342	0.88002292	-
4	0.561214914	0.381559619	-
5	0.558303255	0.208657603	-
6	0.207638495	0.899744752	-
7	0.190312039	0.855341981	-
8	0.676018218	0.462655623	-
9	0.312009526	0.911507006	-
10	0.601710253	0.164577799	-
11	0.115641375	0.966328072	-
12	0.131218144	0.954649164	-
13	0.266678747	0.812043548	-
14	0.318311027	0.864916393	-
15	0.272872551	0.843235922	-
16	0.39804113	0.770547481	-
17	0.081348066	0.996655727	-
18	1	0	-
19	0.280437213	0.761806532	-
20	0	0.859842684	-
21	0.415491692	0.838380997	-
22	0.9617838	0.272761369	-
23	0.242375823	1	-
24	0.449677809	0.872011252	-
25	0.259744637	0.802656665	-
26	0.534877004	0.963567224	-
Max	1	1	**-**

Table 6

Normalization Experimental Results for 36 Units

Runs	Cost	Reliability	Violation
1	0.806659467	0.955622677	-
2	0.123507715	0.392832249	-
3	0.589150411	0.734549257	-
4	0.718656399	0.477375697	-
5	0.544546437	0.969505112	-
6	0.203554256	0	-
7	0.835546179	0.654158922	-
8	0.754032586	1	-
9	0.700512962	0.636355716	-
10	0.270038405	0.479234433	-
11	0.543570407	0.74018355	-
12	0.120652021	0.765741171	-
13	0.21074714	0.325627323	-
14	0.885680843	0.841049024	-
15	0.984033191	0.676579926	-
16	0.744013001	0.492768355	-
17	0.777012835	0.206145446	-
18	0.837953521	0.794057853	-
19	0.898214659	0.742652184	-
20	0	0.526399861	-
21	0.55966564	0.699378485	-
22	0.642033997	0.687732342	-
23	0.284407175	0.207888011	-
24	1	0.444731645	-
25	0.51857709	0.296061803	-
26	0.861636639	0.484142658	-
Max	1	1	**-**

Deviation Sequences

The normalized data for cost, reliability, and violation were then used to calculate the deviation sequence (d_{ij}) using Equation 3 (Jozić et al., 2015):

$$\boldsymbol{d_{ij} = max(x_{ij}) - x_{ij}} \qquad (3)$$

where x_{ij} is the normalization value. Tables 7 and 8 display the deviation sequence for cost, reliability, and violation.

Table 7

Deviation Sequences (d_{ij}) for 26 Units

Runs	Cost	Reliability	Violation
1	0.927293569	0.132812419	-
2	0.551251306	0.269833828	-
3	0.597702658	0.11997708	-
4	0.438785086	0.618440381	-
5	0.441696745	0.791342397	-
6	0.792361505	0.100255248	-
7	0.809687961	0.144658019	-
8	0.323981782	0.537344377	-
9	0.687990474	0.088492994	-
10	0.398289747	0.835422201	-
11	0.884358625	0.033671928	-
12	0.868781856	0.045350836	-
13	0.733321253	0.187956452	-
14	0.681688973	0.135083607	-
15	0.727127449	0.156764078	-
16	0.60195887	0.229452519	-
17	0.918651934	0.003344273	-
18	0	1	-
19	0.719562787	0.238193468	-
20	1	0.140157316	-
21	0.584508308	0.161619003	-
22	0.0382162	0.727238631	-
23	0.757624177	0	-
24	0.550322191	0.127988748	-
25	0.740255363	0.197343335	-
26	0.465122996	0.036432776	-
Min	0	0	-
Max	1	1	-

Table 8

Deviation Sequences (d_{ij}) for 36 Units

Runs	Cost	Reliability	Violation
1	0.193340533	0.044377323	-
2	0.876492285	0.607167751	-
3	0.410849589	0.265450743	-
4	0.281343601	0.522624303	-
5	0.455453563	0.030494888	-
6	0.796445744	1	-
7	0.164453821	0.345841078	-
8	0.245967414	0	-
9	0.299487038	0.363644284	-
10	0.729961595	0.520765567	-
11	0.456429593	0.25981645	-
12	0.879347979	0.234258829	-
13	0.78925286	0.674372677	-
14	0.114319157	0.158950976	-
15	0.015966809	0.323420074	-
16	0.255986999	0.507231645	-
17	0.222987165	0.793854554	-
18	0.162046479	0.205942147	-
19	0.101785341	0.257347816	-
20	1	0.473600139	-
21	0.44033436	0.300621515	-
22	0.357966003	0.312267658	-
23	0.715592825	0.792111989	-
24	0	0.555268355	-
25	0.48142291	0.703938197	-
26	0.138363361	0.515857342	-
Min	**0**	**0**	-
Max	**1**	**1**	-

Gray Relational Coefficients and Gray Relational Grade

The gray relational coefficient (GRC) can be expressed as in Equation 4 (Jozić et al., 2015):

$$\boldsymbol{GRC} = \frac{(\boldsymbol{\Delta_{min}} + \boldsymbol{\xi\Delta_{max}})}{(\boldsymbol{\Delta_{ij}} + \boldsymbol{\xi\Delta_{max}})} \tag{4}$$

where Δ_{ij} is the deviation sequence, Δ_{min} and Δ_{max} are the minimum and maximum values of the deviation sequence, and ξ is the distinguishing coefficient. In this research, the value of ξ was assumed to be 0.5 as in Jozić et al. (2015). The GRG values were calculated after obtaining the GRC values, which can be expressed as in Equation 5 (Jozić et al., 2015):

$$GRG = \frac{1}{n}\sum_{k=1}^{n} GRC \tag{5}$$

In this research, the number of objectives was three. Tables 9 and 10 show the GRC and GRG values for the three systems. In particular, the GRG values in Tables 9 and 10 were obtained by dividing the sum of GRC by 2 because there was no GRC value for violation. The higher values of GRG were preferred (Kolahan & Azadi Moghaddam, 2015).

Table 9

Gray Relational Coefficients and Gray Relational Grade for 26 Units

Runs	GRC (Cost)	GRC (Reliability)	GRC (Violation)	GRG
1	0.350313356	0.790123558	-	0.570218457
2	0.475623666	0.649490814	-	0.56255724
3	0.455496756	0.806481427	-	0.630989092
4	0.532603263	0.447051098	-	0.48982718
5	0.530956492	0.387193978	-	0.459075235
6	0.386888652	0.832978973	-	0.609933812
7	0.381770326	0.775605026	-	0.578687676
8	0.606809532	0.48200001	-	0.544404771
9	0.420878796	0.849627787	-	0.635253291
10	0.556613277	0.374413425	-	0.465513351
11	0.361178087	0.936905192	-	0.649041639
12	0.365288302	0.916840989	-	0.641064646
13	0.405409376	0.72679019	-	0.566099783
14	0.423123183	0.787297916	-	0.60521055
15	0.40745564	0.761308386	-	0.584382013
16	0.453737443	0.685445574	-	0.569591508
17	0.352447269	0.993355894	-	0.672901582
18	1	0.333333333	-	0.666666667
19	0.409982992	0.677329212	-	0.543656102
20	0.333333333	0.781058011	-	0.557195672
21	0.461038423	0.755721945	-	0.608380184
22	0.928994706	0.407418726	-	0.668206716

(continued)

Runs	GRC (Cost)	GRC (Reliability)	GRC (Violation)	GRG
23	0.397575054	1	-	0.698787527
24	0.476044402	0.796192609	-	0.636118506
25	0.403142784	0.717006925	-	0.560074854
26	0.518068684	0.93208324	-	0.725075962

Table 10

Gray Relational Coefficients and Gray Relational Grade for 36 Units

Runs	GRC (Cost)	GRC (Reliability)	GRC (Violation)	GRG
1	0.721146358	0.91848058	-	0.819813469
2	0.363242138	0.451602749	-	0.407422443
3	0.548938053	0.653209895	-	0.601073974
4	0.639923331	0.488938116	-	0.564430724
5	0.523311671	0.94251615	-	0.73291391
6	0.385669822	0.333333333	-	0.359501578
7	0.752497742	0.591127592	-	0.671812667
8	0.670270565	1	-	0.835135283
9	0.625401009	0.578942059	-	0.602171534
10	0.406516758	0.489828435	-	0.448172596
11	0.522777634	0.658053666	-	0.59041565
12	0.362490109	0.680958785	-	0.521724447
13	0.387821517	0.425759224	-	0.406790371
14	0.813909178	0.758781789	-	0.786345483
15	0.969054581	0.607223476	-	0.788139028
16	0.661387035	0.496410138	-	0.578898587
17	0.691575209	0.3864422	-	0.539008705
18	0.755233985	0.708273337	-	0.731753661
19	0.83086105	0.660198642	-	0.745529846
20	0.333333333	0.513557856	-	0.423445595
21	0.531725758	0.624514818	-	0.578120288
22	0.582773675	0.615560641	-	0.599167158
23	0.411321941	0.386963363	-	0.399142652
24	1	0.473813128	-	0.736906564
25	0.509464365	0.41530371	-	0.462384038
26	0.783252972	0.492195094	-	0.637724033

Signal-to-Noise Ratio Analysis

The GRA method was employed to transform all three objectives into a single criterion called GRG. The S/N analysis was then applied to the GRG results to determine the new values for the parameters of the PACS algorithm. The S/N ratio was calculated using Equation 6 (Kolahan & Azadi Moghaddam, 2015):

$$\frac{S}{N} = -10 \log \left(\frac{1}{m} \sum_{i=1}^{m} \frac{1}{GRG^2} \right) \tag{6}$$

where *m* is the number of runs in a trial and GRG is the value obtained as the result of the GRA process. In this research, *m* is equal to 1 as considered by Kolahan and Azadi Moghaddam (2015). Tables 11 and 12 display the results of S/N for the three systems.

Table 11

Response of S/N for 26 Units

Runs	Alpha	Beta	Initial Pheromone	Global Rate	Local Rate	Exploration Probability	S/N
1	0.005	0.005	0.01	0.005	0.005	0.1	-4.87917
2	0.005	0.500	0.10	0.050	0.050	0.6	-4.99667
3	0.005	1.000	0.20	0.100	0.100	0.7	-3.99956
4	0.005	1.500	0.50	0.200	0.200	0.8	-6.19914
5	0.005	2.000	1.50	0.250	0.250	0.9	-6.76232
6	0.500	0.005	0.10	0.100	0.200	0.9	-4.29435
7	0.500	0.500	0.20	0.200	0.250	0.1	-4.75112
8	0.500	1.000	0.50	0.250	0.005	0.6	-5.28156
9	0.500	1.500	1.50	0.005	0.050	0.7	-3.94106
10	0.500	2.000	0.01	0.050	0.100	0.8	-6.64136
11	1.000	0.005	0.20	0.250	0.050	0.8	-3.75455
12	1.000	0.500	0.50	0.005	0.100	0.9	-3.86196
13	1.000	1.000	1.50	0.050	0.200	0.1	-4.94214
14	1.000	1.500	0.01	0.100	0.250	0.6	-4.36187
15	1.000	2.000	0.10	0.200	0.005	0.7	-4.66606
16	1.500	0.005	0.50	0.050	0.250	0.7	-4.88873
17	1.500	0.500	1.50	0.100	0.005	0.8	-3.44097
18	1.500	1.000	0.01	0.200	0.050	0.9	-3.52183
19	1.500	1.500	0.10	0.250	0.100	0.1	-5.29351
20	1.500	2.000	0.20	0.005	0.200	0.6	-5.07985
21	2.000	0.005	1.50	0.200	0.100	0.6	-4.31650
22	2.000	0.500	0.01	0.250	0.200	0.7	-3.50178
23	2.000	1.000	0.10	0.005	0.250	0.8	-3.11310
24	2.000	1.500	0.20	0.050	0.005	0.9	-3.92924
25	2.000	2.000	0.50	0.100	0.050	0.1	-5.03508
26	1.000	0.005	0.01	0.100	0.005	0.1	-2.79233

Table 12

Response of S/N for 36 Units

Runs	Alpha	Beta	Initial pheromone	Global rate	Local rate	Exploration probability	S/N
1	0.005	0.005	0.01	0.005	0.005	0.1	-1.72570
2	0.005	0.500	0.10	0.050	0.050	0.6	-7.79910
3	0.005	1.000	0.20	0.100	0.100	0.7	-4.42144
4	0.005	1.500	0.50	0.200	0.200	0.8	-4.96779
5	0.005	2.000	1.50	0.250	0.250	0.9	-2.69894
6	0.500	0.005	0.10	0.100	0.200	0.9	-8.88598
7	0.500	0.500	0.20	0.200	0.250	0.1	-3.45504
8	0.500	1.000	0.50	0.250	0.005	0.6	-1.56486
9	0.500	1.500	1.50	0.005	0.050	0.7	-4.40560
10	0.500	2.000	0.01	0.050	0.100	0.8	-6.97109
11	1.000	0.005	0.20	0.250	0.050	0.8	-4.57684
12	1.000	0.500	0.50	0.005	0.100	0.9	-5.65118
13	1.000	1.000	1.50	0.050	0.200	0.1	-7.81259
14	1.000	1.500	0.01	0.100	0.250	0.6	-2.08773
15	1.000	2.000	0.10	0.200	0.005	0.7	-2.06794
16	1.500	0.005	0.50	0.050	0.250	0.7	-4.74795
17	1.500	0.500	1.50	0.100	0.005	0.8	-5.36808
18	1.500	1.000	0.01	0.200	0.050	0.9	-2.71270
19	1.500	1.500	0.10	0.250	0.100	0.1	-2.55070
20	1.500	2.000	0.20	0.005	0.200	0.6	-7.46405
21	2.000	0.005	1.50	0.200	0.100	0.6	-4.75964
22	2.000	0.500	0.01	0.250	0.200	0.7	-4.44904
23	2.000	1.000	0.10	0.005	0.250	0.8	-7.97744
24	2.000	1.500	0.20	0.050	0.005	0.9	-2.65175
25	2.000	2.000	0.50	0.100	0.050	0.1	-6.69994
26	1.000	0.005	0.01	0.100	0.005	0.1	-3.90734

Based on the S/N values, the analysis of means was conducted to determine a new combination of parameter values produced from the two systems (i.e., 26- and 36-unit systems). The means of the S/N values at different levels are calculated for each design parameter. A greater S/N corresponds to a better performance (Kolahan & Azadi Moghaddam, 2015). The mean S/N values for the parameters at each level are presented in Tables 13 and 14 for the 26- and 36-unit systems, respectively. The optimal mean S/N values for each parameter were highlighted. The optimal S/N values were used to determine the new parameter values. This can be done by plotting a graph of candidate values against mean S/N values for each parameter as shown in Figures 2 and 3. The new candidate values were those that corresponded to the highest mean of S/N values.

Table 13

Mean of S/N for 26-Unit System

Parameters	Level 1	Level 2	Level 3	Level 4	Level 5
Alpha	-5.367	-4.982	-4.063	-4.445	**-3.979**
Beta	-4.154	**-4.110**	-4.172	-4.745	-5.637
Initial Pheromone	**-4.283**	-4.473	-4.303	-5.053	-4.681
Global Rate	-4.175	-5.080	**-3.987**	-4.691	-4.919
Local Rate	**-4.165**	-4.250	-4.823	-4.803	-4.775
Exploration Probability	-4.616	-4.807	**-4.199**	-4.630	-4.474

Figure 2

Mean of S/N and Candidate Values for 26-Unit System

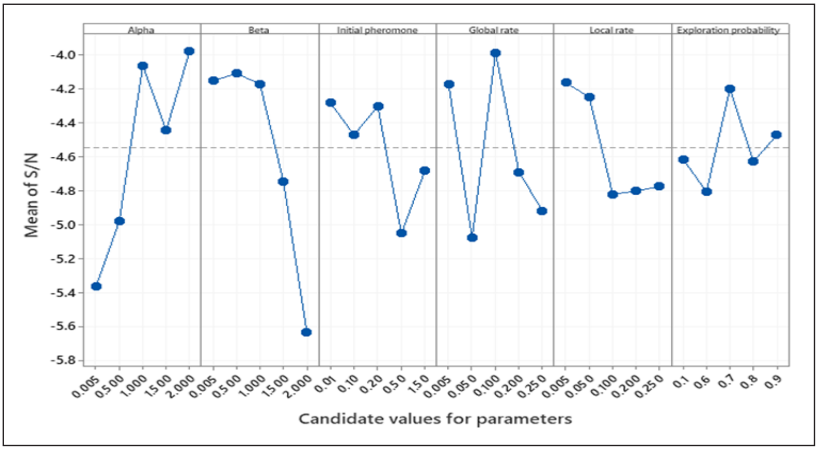


Table 14

Mean of S/N for 36-Unit System

Parameters	Level 1	Level 2	Level 3	Level 4	Level 5
Alpha	**-4.323**	-5.057	-4.351	-4.569	-5.308
Beta	-4.767	-5.344	-4.898	**-3.333**	-5.180
Initial Pheromone	**-3.642**	-5.856	-4.514	-4.726	-5.009
Global Rate	-5.445	-5.996	-5.228	-3.593	**-3.168**
Local Rate	**-2.881**	-5.239	-4.871	-6.716	-4.193
Exploration Probability	-4.359	-4.735	**-4.018**	-5.972	-4.520

Figure 3

Mean of S/N and Candidate Values for 36-Unit System

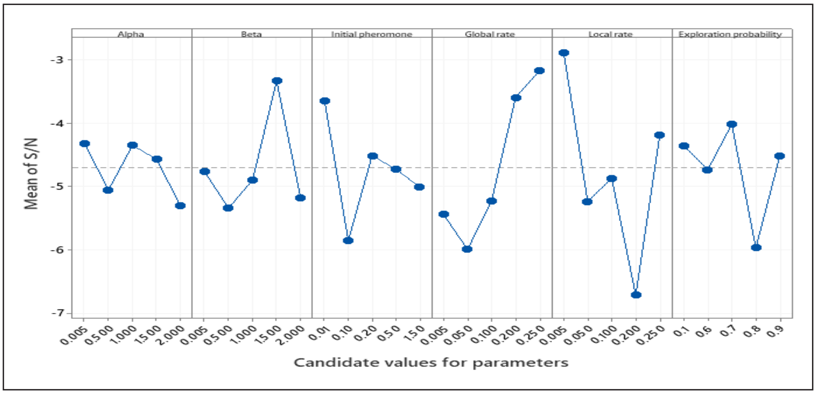


Table 15 presents the summary of the new values of the parameters for the two unit systems. For each parameter, only one new value was used for all three systems in the experiment to test the PACS algorithm. For the Alpha parameter, if Alpha>Beta, the pheromone would guide the algorithm toward solutions with priority given to the objective functions. This also takes into consideration the operation hours of units in deciding their maintenance outage. However, if Beta>Alpha, the heuristic would guide the algorithm toward a solution with priority to the operating hours. This also takes into account the objective functions in deciding the maintenance outage of units. Therefore, for maintenance scheduling of units based on a sequential approach (i.e., operational hours) with priority given to the objective functions, i.e., low cost, high reliability, and low violation, such as in Muthana and Ku-Mahamud (2021; 2022), Alpha>Beta should be chosen. This is supported by the findings in a research by Moncayo-Martínez and Zhang (2011), which obtained better multi-objective optimization results when Alpha>Beta. The finding is also similar to Fattahi et al. (2014), which obtained better single objective optimization results based on operational hours when Alpha>Beta. For this reason, the value of Alpha was set to 2, and the value of Beta was 0.5. Furthermore, the value of Beta was the same as suggested by Berrichi et al. (2010). Fattahi et al. (2014) also chose a small value for Beta. The values for the initial pheromone and the local rate were set to 0.01 and 0.005, respectively since these values were the same for all unit systems. Additionally, Fattahi et al. (2014) selected a smaller value for the global rate, whereas in the current research, the global rate was set to 0.1. A small value for the global rate would always lead to the accumulation of pheromones on the best solution (Fattahi et al., 2014). According to Wang et al. (2015), if Exp<0.5, the algorithm was fond of exploitation. For Exp=0.5, the algorithm had the same probability to perform exploration and exploitation. Nevertheless, if Exp>0.5, the algorithm preferred exploration. Higher exploration might also lead to improved solution quality (Malisia, 2008). Therefore, in this research, the value 0.7 was chosen for the probability of the exploration. The chosen values for the parameters were highlighted.

Table 15

New Parameter Values

Test Systems	Alpha	Beta	Initial Pheromone	Global Rate	Local Rate	Exploration Probability (*Exp*)
26-unit system	**2**	**0.5**	**0.01**	**0.1**	**0.005**	**0.7**
36-unit system	0.005	1.5	**0.01**	0.25	**0.005**	**0.7**

RESULTS AND DISCUSSIONS

Experiments were performed to evaluate the proposed Pareto Ant Colony System I (PACSI) algorithm, which used new values for the parameters. In evaluating the proposed new parameter values, the GRG metric was used. This metric was also applied in Jozić et al. (2015) and Kolahan and Azadi Moghaddam (2015) for evaluating the new parameter values in their multi-objective studies. The PACS benchmark algorithm described in Muthana and Ku-Mahamud (2021; 2022) was used for the comparison. A Friedman test was utilized to show the significant performance of the new parameter values. The experimental results are presented in Table 16 for the objective functions cost, reliability, and violation. In Table 16, PACS1 represented the results of PACS using the new values. In general, it can be seen that the results for all the cost and reliability objective functions were better with the new parameter values for the 26- and 36-unit systems. The results for the violation objective function showed that in all two unit systems, there was no violation in most of the maintenance windows.

Table 16

Results by PACS I and PACS

Test Systems	Algorithms	Maintenance Window	Cost	Reliability	Violation
26-unit system	PACS I	[1000-2000]	**201,316,761.00**	**1,972,341.00**	8
		[1000-2500]	**186,285,879.19**	**1,967,579.00**	0
		[1500-2500]	**186,977,072.81**	**1,985,763.00**	1
		[2000-3000]	**182,373,580.90**	**1,972,386.00**	0
		[2000-4000]	**175,518,540.65**	**2,007,475.00**	0
		[3000-5000]	176,293,493.30	**2,014,789.00**	0
	PACS	[1000-2000]	204,759,986.79	1,970,092.00	8
		[1000-2500]	186,411,157.10	1,957,553.00	0
		[1500-2500]	187,008,609.60	1,970,743.00	**0**
		[2000-3000]	182,654,383.30	1,966,301.00	0
		[2000-4000]	176,354,393.80	1,987,411.00	0
		[3000-5000]	**175,125,928.50**	2,013,313.00	0
36-unit system	PACS I	[1500-2500]	Infeasible	Infeasible	Infeasible
		[1500-3000]	**361,292,184.40**	**2,400,355.00**	0
		[2000-3000]	**362,144,810.77**	2,400,700.00	0
		[2000-4000]	**343,716,769.27**	**2,411,797.00**	0
		[3000-4000]	**344,764,139.14**	**2,415,513.00**	0
		[3000-5000]	**335,912,070.79**	**2,423,701.00**	0
	PACS	[1500-2500]	Infeasible	Infeasible	Infeasible
		[1500-3000]	363,550,376.90	2,397,544.00	0
		[2000-3000]	362,531,811.91	**2,404,129.00**	0
		[2000-4000]	345,982,572.00	2,409,526.00	0
		[3000-4000]	349,295,226.55	2,399,891.00	0
		[3000-5000]	336,980,517.06	2,414,183.00	0

Table 17 shows the GRG improvement results for the PACS algorithm, where the parameters take the benchmark values and the new values. Better PACS performances were obtained with the new values for the 26- and 36-unit systems. In Muthana and Ku-Mahamud (2022), the *p* values of 0.001 and 0.017 were recorded for the 26- and 36-unit systems, respectively, which showed that the performance of these systems can be improved.

Table 17

Gray Relational Grade of PACS I and PACS

Test Systems	Maintenance Window	PACS I (GRG)	PACS (GRG)	GRG Improvement (PACS I, PACS)
26-unit system	[1000-2000]	**1.000**	0.333	**0.667**
	[1000-2500]	**1.000**	0.333	**0.667**
	[1500-2500]	**1.000**	0.333	**0.667**
	[2000-3000]	**1.000**	0.333	**0.667**
	[2000-4000]	**1.000**	0.333	**0.667**
	[3000-5000]	0.667	0.667	0.000
36-unit system	[1500-2500]	Infeasible	Infeasible	Infeasible
	[1500-3000]	**1.000**	0.333	**0.667**
	[2000-3000]	0.667	0.667	0.000
	[2000-4000]	**1.000**	0.333	**0.667**
	[3000-4000]	**1.000**	0.333	**0.667**
	[3000-5000]	**1.000**	0.333	**0.667**

To show the comparison statically, Table 18 summarizes the results obtained by the Friedman test in which the *p* value was used to show if there was a significant difference in performance. The GRG results were also used to calculate the *p* values for the two unit systems. It can be seen that the PACSI algorithm outperformed the PACS algorithm in the 26- and 36-unit systems. The computed *p* values for the 26- and 36-unit systems were less than 0.05, indicated that there was a significant difference in terms of the GRG values between PACSI and PACS. This implied that the PAC algorithm was significantly better when the new parameter values were used compared to the benchmark values. The 36-unit system had a bigger *p* value than the 26-unit system because of bigger demand and system size, which increased the problem's complexity. Therefore, it was difficult for the algorithm to obtain better solutions.

Table 18

Results of Friedman Test

Test Systems	Algorithms	Mean Rank	Ranking
	PACS I	1.92	1
26-unit system	PACS	1.08	2
	P value	**0.025**	
	PACS I	1.90	1
36-unit system	PACS	1.10	2
	P value	**0.046**	

CONCLUSION

This research proposed the optimal parameter values for the multi-objective PACS algorithm. The Taguchi-GRA method was used to obtain optimal values for the algorithm's parameters. Performance evaluation of the proposed new parameter values for the tested systems showed that the multi-objective PACSI algorithm was able to obtain better GRG solutions. Therefore, the algorithm was able to achieve better maintenance scheduling with low cost, high reliability, and low violation. The obtained new values for the parameters can be used as benchmark values in solving multi-objective GMS problems using the multi-objective PACS algorithm and its variants.

Future research could focus on the weights of the three parameters that represent the cost, reliability, and violation. These weights can be tested with fixed values to make a comparison between the random and fixed values in providing a better maintenance schedule. In addition, the proposed research has only used a single value for all the constants attached to the three objective functions, i.e., the constant with value one was used. The constants were used in deciding the best amount of pheromone in the reward to the best solution (solution with low cost, high reliability, and low violation) for the global update process in the end of every iteration. Other values can be tested to find the best value for the constant, which will determine the amount of pheromone to be used for the global update.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

Ankita & Sahana, S. K. (2019). An automated parameter tuning method for ant colony optimization for scheduling jobs in grid environment. *International Journal of Intelligent Systems and Applications*, *11*(3), 11–21. https://doi.org/10.5815/ijisa.2019.03.02

Ariyaratne, M. K. A., & Fernando, T. G. I. (2018). A self-tuning firefly algorithm to tune the parameters of ant colony system. *International Journal of Swarm Intelligence*, *3*(4), 309–331. https://doi.org/10.1080/09692290.2013.878741

Berrichi, A., Yalaoui, F., Amodeo, L., & Mezghiche, M. (2010). Bi-objective ant colony optimization approach to optimize production and maintenance scheduling. *Computers and Operations Research*, *37*(9), 1584–1596. https://doi.org/10.1016/j.cor.2009.11.017

Chagas, J. B. C., & Wagner, M. (2020). Ants can orienteer a thief in their robbery. *Operations Research Letters*, *48*(6), 708–714. https://doi.org/10.1016/j.orl.2020.08.011

Chagas, J. B. C., & Wagner, M. (2022). Efficiently solving the thief orienteering problem with a max–min ant colony optimization approach. *Optimization Letters*, *16*(8), 2313–2331. https://doi.org/10.1007/s11590-021-01824-y

Drozdik, M., Aguirre, H., Akimoto, Y., & Tanaka, K. (2015, January). Comparison of parameter control mechanisms in multi-objective differential evolution. In *International Conference on Learning and Intelligent Optimization (LION 2015)* (pp. 89–103). https://doi.org/10.1007/978-3-319-19084-6_8

Fattahi, M., Mahootchi, M., Mosadegh, H., & Fallahi, F. (2014). A new approach for maintenance scheduling of generating units in electrical power systems based on their operational hours. *Computers and Operations Research*, *50*(Feb 1), 61–79. https://doi.org/10.1016/j.cor.2014.04.004

Förster, M., Bickel, B., Hardung, B., & Gabriella, K. (2007, July). Self-adaptive ant colony optimisation applied to function allocation in vehicle networks. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation* (pp. 1991–1998). https://dl.acm.org/doi/pdf/10.1145/1276958.1277352?casa_token=SORrdgdp36IAAAAA:IsoPtCQUPA865C4pnTx5GI3UoXTnJNzwVxQIn0afFcR8Uv-mFV9_SEwa5PLUyDEopBUF6sQdKXCcww

Han, Z., Wang, Y., & Tian, D. (2021). Ant colony optimization for assembly sequence planning based on parameters optimization. *Frontiers of Mechanical Engineering*, *16*, 393–409. https://doi.org/10.1016/j.asoc.2021.107439

Jozić, S., Bajić, D., & Celent, L. (2015). Application of compressed cold air cooling: Achieving multiple performance characteristics in end milling process. *Journal of Cleaner Production*, *100*(Aug 1), 325–332. https://doi.org/10.1016/j.jclepro.2015.03.095

Kolahan, F., & Azadi Moghaddam, M. (2015). The use of Taguchi method with grey relational analysis to optimize the EDM process parameters with multiple quality characteristics. *Scientia Iranica*, *22*(2), 530–538. http://scientiairanica.sharif.edu/article_3665_0e7b4362f86839d882e60aa681d5ce56.pdf

Lezama, F., Faia, R., Soares, J., Faria, P., & Vale, Z. (2020, July). Learning bidding strategies in local electricity markets using ant colony optimization. In *IEEE Congress on Evolutionary Computation (CEC)* (pp. 1–8).

López-Ibáñez, M., Stützle, T., & Dorigo, M. (2015). Ant colony optimization: A component-wise overview. In *Handbook of heuristics* (pp. 371-407). Springer, Cham

Lyu, Z., Wang, Z., Duan, D., Lin, L., Li, J., Yang, Y., Chen, Y., & Li, Y. (2020). Tilting path optimization of tilt quad rotor in conversion process based on ant colony optimization algorithm. *IEEE Access*, *8*(July), 140777–140791. https://doi.org/10.1109/ACCESS.2020.3012672

Malisia, A. R. (2008). Improving the exploration ability of ant-based algorithms. In *Oppositional Concepts in Computational Intelligence* (pp. 121–142). https://link.springer.com/content/pdf/10.1007/978-3-540-70829-2.pdf

Manikandan, N., Kumanan, S., & Sathiyanarayanan, C. (2015). Multi response optimization of electrochemical drilling of titanium Ti6Al4V alloy using Taguchi based grey relational analysis. *Indian Journal of Engineering and Materials Sciences*, *22*(April), 153–160.

Mavrovouniotis, M., Ioannou, A., & Yang, S. (2017, March). Pre-scheduled colony size variation in dynamic environments. In *European Conference on the Applications of Evolutionary Computation* (pp. 128–139). https://doi.org/10.1007/978-3-319-55792-2_9

Moncayo-Martínez, L. A., & Zhang, D. Z. (2011). Multi-objective ant colony optimisation: A meta-heuristic approach to supply chain design. *International Journal of Production Economics*, *131*(1), 407–420. https://doi.org/10.1016/j.ijpe.2010.11.026

Muthana, S. A., & Ku-Mahamud, K. R. (2022). Tri-objective generator maintenance scheduling model based on sequential strategy. *PLOS ONE*, *17*(10), 1–29. https://doi.org/10.1371/journal.pone.0276225

Muthana, S. A., & Ku-Mahamud, K. R. (2021). Generator Maintenance scheduling models for electrical power systems: A review. *International Journal of Electrical and Electronic Engineering & Telecommunications*, *10*(5), 307–318. http://www.ijeetc.com/uploadfile/2021/0408/20210408102935567.pdf

Muthana, S. A., & Ku-Mahamud, K. R. (2022). Comparison of multi-objective optimization methods for generator maintenance scheduling. *International Journal of Intelligent Engineering and Systems*, *15*(4), 599–614. https://doi.org/10.22266/ijies2022.0831.54

Negulescu, A. E. (2017). Normalization of ACO algorithm parameters. *University Politehnica Bucharest*, *79*(2), 71–82.

Ragmani, A., Elomri, A., Abghour, N., Moussaid, K., & Rida, M. (2019). An improved hybrid fuzzy-ant colony algorithm applied to load balancing in cloud computing environment. *Procedia Computer Science*, *151*(2019), 519–526. https://doi.org/10.1016/j.procs.2019.04.070

Ragmani, A., Elomri, A., Abghour, N., Moussaid, K., & Rida, M. (2020). FACO: A hybrid fuzzy ant colony optimization algorithm for virtual machine scheduling in high-performance cloud computing. *Journal of Ambient Intelligence and Humanized Computing*, *11*(10), 3975–3987. https://doi.org/10.1007/s12652-019-01631-5

Sagban, R. (2016). *Reactive approach for automating exploration and exploitation in ant colony optimization* [Doctoral dissertation, Universiti Utara Malaysia]. https://www.researchgate.net/publication/303446764_Reactive_Approach_for_Automating_Exploration_and_Exploitation_in_Ant_Colony_Optimization

Sihem, B., & Benmansour, A. (2018). Ant colony parameters of electric power system reliability by Taguchi method. *EEA - Electrotehnica, Electronica, Automatica*, *66*(2), 57–60.https://www.researchgate.net/profile/Sihem-Bouri/

publication/340383187_Ant_colony_parameters_of_electric_power_system_reliability_by_Taguchi_method/links/5e85db33a6fdcca789e965c4/Ant-colony-parameters-of-electric-power-system-reliability-by-Taguchi-method.pdf?_sg%5B0%5D=started_experiment_milestone&origin=journalDetail

Talbi, E.-G. (2009). *Metaheuristics: From design to implementation*. John Wiley & Sons. https://scholar.google.com.my/scholar?q=Metaheuristics:+From+design+to+implementation&hl=en&as_sdt=0&as_vis=1&oi=scholart

Tirkolaee, E. B., Alinaghian, M., Hosseinabadi, A. A. R., Sasi, M. B., & Sangaiah, A. K. (2019). An improved ant colony optimization for the multi-trip capacitated Arc routing problem. *Computers and Electrical Engineering*, *77*(Jul 1), 457–470. https://doi.org/10.1016/j.compeleceng.2018.01.040

Vinay, V. P., & Sridharan, R. (2013). Taguchi method for parameter design in ACO algorithm for distribution-allocation in a two-stage supply chain. *International Journal of Advanced Manufacturing Technology*, *64*(Feb 1), 1333–1343. https://doi.org/10.1007/s00170-012-4104-5

Wang, G., Chu, H. C. E., Zhang, Y., Chen, H., Hu, W., Li, Y., & Peng, X. J. (2015). Multiple parameter control for ant colony optimization applied to feature selection problem. *Neural Computing and Applications*, *26*(7), 1693–1708. https://doi.org/10.1007/s00521-015-1829-8

Wang, Y., & Han, Z. (2021). Ant colony optimization for traveling salesman problem based on parameters optimization. *Applied Soft Computing*, *107*(Aug), 107439. https://doi.org/10.1016/j.asoc.2021.107439

Yasear, S. A., & Ku-Mahamud, K. R. (2021). Review of multi-objective swarm intelligence optimization algorithms. *Journal of Information and Communication Technology*, *20*(2), 171–211. https://doi.org/10.32890/jict2021.20.2.3

Yuan-Kang, W., Chih-Cheng, H., & Chun-Liang, L. (2013). Resolution of the unit commitment problems by using the hybrid Taguchi-ant colony system algorithm. *International Journal of Electrical Power and Energy Systems*, *49*(Jul 1), 188–198. https://doi.org/10.1016/j.ijepes.2013.01.007

Zheng, F., Zecchin, A. C., Newman, J. P., Maier, H. R., & Dandy, G. C. (2017). An adaptive convergence-trajectory controlled ant colony optimization algorithm with application to water distribution system design problems. *IEEE Transactions on Evolutionary Computation*, *21*(5), 773–791. https://doi.org/10.1109/TEVC.2017.2682899

Zhou, R., Lee, H. P., & Nee, A. Y. C. (2008). Applying ant colony optimisation (ACO) algorithm to dynamic job shop scheduling problems. *International Journal of Manufacturing Research*, *3*(3), 301. https://doi.org/10.1504/IJMR.2008.019212