Received: March 28, 2022. Revised: June 15, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Comparison of Multi-objective Optimization Methods for Generator Maintenance Scheduling

Shatha Abdulhadi Muthana[1]* **Ku Ruhana Ku-Mahamud[2,3]**

[1]*General Company of South Electricity Distribution, Ministry of Electricity, Iraq*
[2]*School of Computing, Universiti Utara Malaysia, Malaysia*
[3]*Shibaura Institute of Technology, Tokyo, Japan*
* Corresponding author's Email: shatha_muthana@yahoo.com

Abstract: This study presents a comparison between multi-objective optimization methods used to obtain solution for generator maintenance scheduling (GMS) problem. The GMS problem with three objectives which include the total operation cost minimization, system's reliability (gross reserve) maximization and convenience is considered in this study. Convenience objective is represented by the minimization of the violation in the number of units' maintenance outage constraint. To solve the problem of GMS, there is a need for GMS model to represent the requirements of electrical power system and optimization method to implement the model and obtain solution for the GMS. The proposed Pareto ant colony system (PACS) algorithm is based on ant colony system algorithm and Pareto approach. Pareto approach is used to make trade-off between the obtained solutions based on the three objectives. In this study comparison is made based on the results of experiments using the IEEE RTS 32- and 36-unit systems while the demand for the 32- and 36-unit systems is based on IEEE RTS demand systems. In addition, four common multi-objective algorithms based on Pareto approach i.e., the Non-dominated Sorting Genetic Algorithm II, Strength Pareto Evolutionary Algorithm 2, Multi-objective Simulated Annealing, and Multi-objective Particle Swarm Optimization are used in the evaluation of the proposed PACS algorithm. The multi-objective GMS model is implemented by all the algorithms and five performance metrics i.e., the grey relational grade (GRG), coverage, distance to Pareto front, overall Pareto spread and the number of non-dominated solutions are used in the evaluation. The Friedman test is also used to evaluate the algorithms' performance statically, which is made based on GRG metric. The experimental results showed that the proposed PACS algorithm was able to obtain a robust solution by considering different initial operational hours of the units. In term of GRG metric, the PACS algorithm was able to obtain the best results for the 32-unit systems in all the maintenance windows. However, for the 36-unit system, the PACS algorithm secured the second-best results at the early stages of the operation time but outperformed other algorithms during other operation times. For other metrics, overall the PACS algorithm has the best performance in terms of coverage, distance to Pareto front and overall Pareto spread metrics while the NSGAII has the best result in terms of the number of obtained non-dominated solutions. Friedman test implies that the PACS algorithm is significantly better than the other comparative algorithms.

Keywords: Optimization, Pareto approach, Scheduling, Generator maintenance, Multi-objective.

1. Introduction

Generator maintenance scheduling (GMS) problem is a complex and nonlinear optimization problem that specifies the schedule for carrying out planned preventive maintenance on power generation units [1]. To solve the problem of GMS, there is a need for a model and optimization method. Various generator maintenance scheduling models were developed to represent the single objective [2-8] and multi-objective [9-11] electrical power system requirements to generate schedules for maintaining generating units. Optimization methods are required to obtain solutions for any GMS. There are two types of optimization methods: single objective and multi-objective optimization methods.

Single objective optimization methods aim to obtain solution for the single objective problems [1]. Multi-objective optimization problems aim to uncover the best compromise between conflicting and multiple objectives [12]. Thus, due to the need of simultaneous optimization of several objectives at that single moment, and the difficulty and complications in providing a solution, the multi-objective optimization methods are often experienced by researchers [12, 13]. Multi-objective optimization methods are classified as classical methods (aggregative methods) and intelligent methods (multi-objective metaheuristics) [14, 15]. Aggregative methods (i.e., classical methods) convert the multi-objective problem into a single objective problem either through creating an aggregate objective function or via an optimization of one objective and considering the other as a constraint [10, 16]. These methods are pushed towards uncovering only one solution and are essential in instances in which preferential information relating to the objectives is clearly understood in advance. The most generic methods in the aggregation approach are the weighted sum, and the ε-constraint methods [14]. The main drawback of classical methods is that the user must execute several runs through different parameter settings for the purpose of generating a representative approximation of the overall Pareto front (i.e., optimal solutions) [14]. Furthermore, transforming the multi-objective problem into a single objective problem cannot achieved the feasibility of trade-offs among multi-objectives [17].

Intelligent methods which have been developed for solving multi-objective problems simultaneously are in contrast to aggregation methods. Aggregation methods are reasonably straightforward, and no modifications are required for the basic algorithm [16]. The most common intelligent methods that are used in the domain of multi-objective GMS are the Non-dominated Sorting Genetic Algorithm II (NSGAII), Strength Pareto Evolutionary Algorithm2 (SPEA2), Pareto Ant Colony Optimization, Multi-objective Simulated Annealing (MOSA), and Multi-objective Particle Swarm Optimization (MOPSO) [9, 18-26].

Based on the literature [27, 28], classical methods are considered to be non-Pareto based techniques, while the intelligent methods are considered to be Pareto based techniques. The basic idea of Pareto based techniques is that Pareto front is directly incorporated into the concept of Pareto optimum. Non-Pareto based techniques are approaches that do not directly incorporate the concept of Pareto optimum [27]. In addition, [29] give a classification to algorithms depending on whether the output of the algorithms is a set of Pareto solutions (i.e., a set of optimal solutions) or the output is a single solution. The Pareto dominance approach uses the Pareto dominance relation to select non-dominated solutions. According to the Pareto dominance relation, the dominance concept can be expressed as: A solution X1 can be said to have dominance over another solution X2 if both conditions, as follows, are true: (i) solution X1 is not considered worse than X2 in the entire objectives; and (ii) solution X1 is considered a better solution than X2 in one of the objectives at least [15]. In the case of conflicting objectives in multi-objective optimization, Pareto efficiency or Pareto optimality are ideal solutions. The Pareto space is considered by most authors [15]. Securing a set of Pareto-optimal solutions supports the decision maker with further comprehensive understanding of the whole feasible solutions to enable an acceptable final plan for GMS compared to one single optimal solution [10, 13, 14, 30].

This paper presents a comparison of five multi-objective optimization algorithms for GMS for a multi-objective GMS model described in [1]. The novelty of this paper lies in demonstrating the proposed multi-objective PACS algorithm in obtaining better solution for the multi-objective GMS problem with strategy based on operational hours, compared to other multi-objective optimization methods (i.e., NSGAII, SPEA2, MOSA, and MOPSO). New results based on different metrices provided in this paper have proved the effectiveness of the proposed algorithm. Section 2 discusses previous studies on multi-objective optimization methods. This is followed by Section 3 which describes the proposed optimization method. In Section 4 experimental design to evaluate the performance of the proposed method is presented and Section 5 describes the results of the experiments. Finally, Section 6 depicts the conclusions and future work.

2. Related literature

The studies of [1, 9-11, 18, 20, 26] consider the approach of Pareto based techniques which has proved its efficiency to get better solutions in optimization of the multi-objective problems. Moreover, all the optimization methods are developed to solve one problem with multi-objectives, except the study in [1], which was developed to solve multi problems with multi-objectives.

Received: March 28, 2022. Revised: June 15, 2022.

Table 1 summarizes the most recent studies of solution methods in multi-objective optimization that are used to obtain solutions for multi-objective GMS problem. The study of [10] consider optimization method to obtain solution for GMS with three types of objective functions; comprising producer profit, system reliability which is defined as the minimization of the standard deviation of the reliability index, and total generation cost. The study of [20] considers optimization method to obtain solution to GMS with two objectives i.e., profit minimization and reliability maximization. The system reliability objective is defined to be the average value of the reliability index. In [9], an optimization method for two objective functions (i.e., maximizing reliability by minimizing the sum of squared reserves and minimizing cost through electricity production cost) has been proposed. The optimization method considered by [11] obtained solution for GMS with the objectives of minimizing the overall operational cost and maximizing the deterministic reliability of the power system by maximizing the average value of reliability index in the planning period. The study in [26] considered optimization method to obtain solution with two objectives comprising maximizing reliability through maximizing the expected rate of energy and minimizing cost through minimizing the total expected costs related to maintenance efforts. In [18] considered optimization method that obtained solution with objectives of cost minimization through maintenance cost. Furthermore, reliability is considered by providing adequate power reserve and the sustainability impacts on a wind farm system. The study in [1] considered optimization method to obtain solution for GMS with the objectives of total operation cost minimization, system's reliability maximization and minimizing the violation in the number of units that send for maintenance outage. Despite all the mentioned studies considered optimization methods to implement multi-objective GMS models and obtained solution for GMS in form of multi-objective. However, all these studies have considered optimization methods to implement unit commitment problem in conjunction with unit maintenance scheduling problem.

Unit commitment are units which are not schedule for maintenance. The unit commitment problem seeks to determine which units are either, i) connected to the power generation system or ii) not connected to the power generation system. This is different from the problem of GMS which seeks to determine which units that are scheduled for maintenance that should enter the maintenance outage. The presented solutions from [9-11, 18, 20, 26] are not efficient enough because unit commitment does not undergo maintenance and have to be solved separately to get more efficient maintenance scheduling of generators. In addition, all the optimization methods did not consider operational hours strategy, where this strategy can extend the lifespan of generating units and can be applied with different types of generating units such as gas turbine [4].

The studies of [9-11, 18, 20] have considered periodic strategy for maintenance scheduling since this strategy is convenient to implement. However, the sequential strategy is more appropriate when the system requires frequent maintenance as it ages [4, 31, 32]. The study of [26] considers sequential maintenance strategy but the age of the unit is calculated as a proportion of the time the unit has been in the system. This does not reflect the actual working time of the units. The method that has been adopted in [4] in deciding the age reflects the exact working hours of the generating unit. Thus, in our previous work [1] we developed the work of [4] and proposed PACS algorithm to be suitable in implementing multi-objective GMS model and produce solution for the GMS with multi-objective.

In [1], a multi-objective PACS algorithm is presented which considered GMS and unit commitment problems separately to be solved simultaneously. The obtained solution method can provide more efficient maintenance scheduling represent the requirements of electrical power system. The Pareto strategy has been employed in these studies because of its efficiency in solving multi-objective problems.

3. The proposed method

The proposed PACS algorithm for the multi-objective GMS scheduling problem adopts and adapts the structure of the single objective ACS algorithm presented by [4]. The proposed algorithm with the scheduling problem explained in our previous work [1].

There are eight constraints in the scheduling problem include GMS constraints, unit commitment constraints and the coupling constraints between GMS and unit commitment. The GMS constraints are maintenance outage units, and continuous maintenance, whilst unit commitment constraints are load balance, minimum system reserve, minimum and maximum capacity of generating units, and minimum up and down time constraints. However, the coupling constraints require participation in an activity of maintenance scheduling between unit commitment and GMS which comprise maintenance

Table 1. Multi-objective solution methods for GMS models

Reference	Pareto-based technique approach	Single problem	Multi problem	Optimization methods
[10]	√	√	X	Group search optimizer with multiple producers.
[9]	√	√	X	Dominance-based multi-objective Simulated Annealing algorithm.
[11]	√	√	X	Fuzzy clustered multi-objective hybrid Differential Evolution algorithm.
[18]	√	√	X	Non-dominated Sorting Genetic Algorithm II.
[20]	√	√	X	Non-dominated Sorting Genetic Algorithm II.
[26]	√	√	X	Multi-objective Particle Swarm Optimization.
[1]	√	X	√	Pareto ant colony system.

online status, as well as maintenance windows. The objectives of the scheduling problem are minimizing the operation cost, maximizing the system reliability, and minimizing the number of violations in maintenance outage units' constraint.

The proposed PACS algorithm is an enhancement of the algorithm presented in [4]. The main difference is the inclusion of more than one objective function, and operational hours are calculated by two factors; the operating hours and start-up times as compared to [4], it is calculated as the accumulated operating hours which neglects the engine start-up times. Inventors have discovered that one engine start is equivalent to 10 hours of operation in terms of the impact on the life of the engine [31]. In addition, the enhancement includes the Pareto approach to make a trade-off between the objectives. The functions for the local and global pheromone updates have also been changed to cater for the multiple objectives.

Table 2 displays the symbols and variables that are used in the pseudo codes.

The pseudo code of the proposed PACS algorithm is as follows:

For each $\boldsymbol{e} \in \boldsymbol{E}$
    Step1. **For** each $\boldsymbol{g} \in \boldsymbol{G}$ **(construct maintenance scheduling solution)**
        1. **For** each $\boldsymbol{j} \in \boldsymbol{J}$
            i. Decide maintenance outage of units according to PACS rules; /*refer Section 3.1*/
            ii. Implement unit commitment heuristic; /*refer Section 3.2*/
            iii. Calculate amount of production for online units /*refer Section 3.3*/
        Endfor
        2. Calculate objective functions;
        3. Update best so far solution;
        4.Perform local pheromone update for ants' group;
        5. If g>1
            Apply Pareto approach;
    Endfor
    Step2. Perform global pheromone update for best ants' groups;
Endfor
Record Pareto front**.**

The algorithm starts with the iteration (e) and in each iteration, there are two processes (i.e., 2 steps). Step 1 is to generate the maintenance scheduling of the generators followed by the update of the global pheromone (in step 2). There are five tasks in step 1 for each ants' group (g), The first task determines which unit goes into which period (j) for maintenance using the PACS algorithm. PACS rules determine what unit should be on maintenance in every period (week). It is assumed, in this research, that the maintenance duration is two periods (i.e., 2 weeks) and is identical for all units in the generated test systems. Next, the algorithm will solve the unit commitment problem. In each period, the unit commitment problem is solved when it satisfies the

Received: March 28, 2022. Revised: June 15, 2022.

Table 2. Nomenclature

Symbol	Meaning
E	Set of main iterations.
G	Set of ant groups.
R	Set of offline units that are free to be online according to the downtime constraint.
F	Set of offline units that are forced to be offline according to the downtime constraint.
O	Set of units out for maintenance without force (i.e., the operational hours of these units are between the lower and upper maintenance window endpoints, however, there is at least one week, 168 h, until the upper endpoint of the window).
Exp	Exploration rate.
φ	Indicator variable that is 1 if the problem is feasible.
i	Index of generating units.
j	Index of periods (weeks).
t	Index of hours.
I	Set of generating units.
J	Set of periods (weeks).
T	Set of hours.
LT	Total hours in each period (168 h in a week).
$pend_{i,j}$	Operational hours of unit *i* at the beginning of period *j* after the last maintenance outage if the outage is started in period *j*; otherwise, it equals to zero.
oph_i^{maxav}	Maximum available operational hours for unit *i* between the last maintenance outage and the next one.
oph_i^{minrq}	Minimum required operational hours for unit *i* between the last maintenance outage and the next one.
N_i	Maximum number of units for maintenance outages in period *j*.
D_i	Duration (of periods) of the maintenance outage for unit *i*.
p_{ijt}	Power generation dispatch of unit *i* in period *j* and time *t*.
D_{jt}	System demand in period *j* and time *t*.
R_{jt}^{min}	Minimum reserve requirement in period *j* and time *t*.
P_i^{min}	Minimum capacity of unit *i*.
P_i^{max}	Maximum capacity of unit *i*.
G_{ijt}	Power generating capacity of unit *i* during period *j* and time *t*.
C_i^P	Production cost ($/MWh) of unit *i*.

load demand by determining the operation schedule of the generating units at every hour interval with varying loads at the lowest production cost (i.e., unit commitment responsible to determine the status of On/Off units outside the maintenance outage). However, the obtained solution might be infeasible in terms of the reserve constraint. Therefore, to make it feasible, four feasibility rules are developed. The algorithm then calculates the production amount for online unit so that the load balance constraint is satisfied. The purpose of this constraint is to ensure the load demand equals the total power production of units in each period and each hour.

The second task is to calculate the objective function after an ants' group has constructed a GMS. This is followed by the updating of the best so far solution (in the third task) and the task of updating the local pheromone for the ants' group is performed. The final task is to check if the there is more than 1 ants' group, the Pareto approach is applied to make a trade-off between the obtained solutions from the ants' groups. After the all the ants' groups in each iteration are completed, the global pheromone update is performed to the best ants' groups. When the iterations are completed, the last Pareto front is recorded for the best GMS solutions.

The following sections describe how to determine the maintenance outage, implementation of the unit commitment heuristic and calculation of the production.

3.1 Maintenance outage determination

An algorithm has been developed to determine the maintenance outage as follows:

While (I ≠ Φ)

Step 1. Choose unit ***i*** with highest $\frac{pend_{i,j}}{oph_i^{maxav}}$ from **I**;

Step 2. **If** ($pend_{i,j} \geq oph_i^{maxav} - LT$)

Perform unit maintenance outage for this period;

Else if

(number of units in maintenance $<$ N_i **and** $pend_{i,j} \geq oph_i^{minrq}$)

1. Compute two heuristic values and *Pr* from unit ***i*** according to Equations (1-3);
2. Generate random number r_1 between (0,1);
3. **If** ($r_1 < Exp$)
   - i. Generate another random number r_2 between (0,1);
   - ii. **If** ($r_2 < Pr$)
     - a. Perform unit maintenance outage for this period;
     - b. Put ***i*** in **O**;

else

 DOI: 10.22266/ijies2022.0831.54

Received: March 28, 2022. Revised: June 15, 2022.

Do not perform unit maintenance outage for this period;

Else if (*Pr > 0.5)*
i. Perform unit maintenance outage for this period;
ii. Put ***i*** in **O**;
Else
Do not perform unit maintenance outage for this period;
Endif
Else
Do not perform unit maintenance outage for this period;
Endif
Endif
Step 3. Take unit ***i*** out from set **I**;
End (while)

In this maintenance outage determination algorithm, PACS rules make decisions about the maintenance outage of each generating unit in three major steps. The GMS and coupling constraints between unit commitment and GMS are considered in this algorithm. Eqs. (1) and (2) are used to calculate the two heuristic values in the probability of unit maintenance outage used in Eq. (3). The probability of the outage increases if operational hours without maintenance for each unit is near to the upper endpoint of the maintenance window. In addition, pheromone values in Eq. (3) are obtained from the updated trail information during previous steps. Three pheromone values are defined where one pheromone is for each objective. The decision of *YES* for pheromone and heuristic in Eq. (3) means the unit should enter maintenance outage, On the contrary, the *NO* indicates not to enter the maintenance outage.

$$\eta_{yes} = 1 - \frac{oph_i^{maxav} - pend_{i,j}}{oph_i^{maxav} - oph_i^{minrq}} \tag{1}$$

$$\eta_{no} = \frac{oph_i^{maxav} - pend_{i,j}}{oph_i^{maxav} - oph_i^{minrq}} \tag{2}$$

$$Pr = \frac{(A_{yes})^{\alpha}.[\eta_{yes}]^{\beta}}{(A_{yes})^{\alpha}.[\eta_{yes}]^{\beta} + (A_{no})^{\alpha}.[\eta_{no}]^{\beta}} \tag{3}$$

where,

$$A_{yes} = [C.\tau_{yes}^{c}] + [R.\tau_{yes}^{r}] + [V.\tau_{yes}^{v}]$$

$$A_{no} = [C.\tau_{no}^{c}] + [R.\tau_{no}^{r}] + [V.\tau_{no}^{v}]$$

3.2 Unit commitment heuristic

The unit commitment heuristic should be implemented for each period in the PACS algorithm. The algorithm for the unit commitment heuristic comprises two major steps. In the first step, for every hour (t) of the present period, the scheduling used for turning the units on or off by considering minimum up-time and down-time constraints. In the second step of the unit commitment heuristic, some of the developed heuristic approaches, named feasibility rules, are implemented to make the acquired solution feasible if reserve constraint of the acquired solution in the first step is not satisfied. The pseudo code of unit commitment heuristic algorithm is as follows:

For each $\boldsymbol{t} \in \boldsymbol{T}$
Step1. **For** each $\boldsymbol{i} \in \boldsymbol{I}$
Decide whether unit should be on or off; /*refer subsection 3.2.1/*
Step2. **If** (unit commitment is infeasible)
Implement feasibility rules; /*refer subsection 3.2.2*/
Endfor

3.2.1. On/Off units' status determination

The heuristic of status determination is relevant to unit commitment decisions in which the up-time and down-time constraints are recognized. The major task here is to get a selection whether units have to be on or off at the present period and hour. Corresponding to this portion of the solution algorithm, at the earlier hour, if the online unit fulfils the up-time constraint and there is no need for it to continue working, the unit turns off by default to minimize the total operational cost. The free units to be turned on, which satisfy the down-time constraint, are maintained in set **R**. The pseudo code for the status determination algorithm is presented in as follows:

If (unit is on maintenance outage)
Let unit be off at the current hour;
Else if (online unit at previous hour does not satisfy up-time constraint)
Let unit be online at current hour;
Else if (offline unit at the previous hour does not satisfy the down-time constraint)
Let the unit be off at the current hour and put ***i*** in **F**;
Else
a.Turn the unit off; /default setting/
b. Put ***i*** in **R**

Endif
Endif
Endif

3.2.2. Feasibility rules

A total of four feasibility rules are developed to make the acquired solution feasible in terms of the reserve constraint. The respective pseudo codes for the rules are illustrated in the following.

Corresponding to the first rule, enough units from set **R** have to be turned on until the reserve constraint is fulfilled. In this case, units with the lowest production cost are selected to be turn on. The pseudo code for the first rule is as follows:

Feasibility rule #1
If (φ = 0)
  While (R≠ Φ)
    If ($\sum_{i\in(onunits)} \mathrm{P}_i^{max} < D_{jt} + R_{jt}^{min}$)
      a. Choose unit *i* with lowest production cost from R;
      b. Turn on unit ***i*** and take unit ***i*** out from set **R**;
    Else
      a. φ ← 1;
      b. No more unit to be turned on
    Endif
  Endwhile
Endif

If the acquired solution does not fulfil the reserve constraint and set **R** is clear, the second feasibility rule must be performed. The second rule is to return the units which are scheduled for maintenance for the present period without obligation to be online. The priority is to return the units is based on the identified criterion expressed in the first task of the while loop in following pseudo code.

Feasibility rule #2
If (φ = 0)
  While (O≠ Φ)
    a. Choose the unit ***i*** with the lowest $\frac{\boldsymbol{c_i^p}}{\boldsymbol{p_i^{max}}\times(oph_i^{maxav}-pend_{i,j})}$ from **O**;
    b. Do not perform unit maintenance outage at this period and return unit ***i*** to the power system;
    c. Turn it on to be available at the current hour by considering the down-time constraint;
    d. Take unit ***i*** out from set **O**;
    e. **If** ($\sum_{i\in(onunits)} \mathrm{P}_i^{max} > D_{jt} + R_{jt}^{min}$)
      i. φ ← 1;
      ii. No more unit to be turned on
    Endif
  Endwhile
Endif

The third feasibility rule will be considered if infeasible solution still occurs because of not fulfilling the reserve constraint. The units in set **F** (those that are obliged to be offline according to the down-time constraint) are probable candidates to be turned on to make the solution feasible. The chosen unit have to be turned on from the earlier hours on which it was turned off by default. For example, if a unit is off for 2 hours, then it must be off for another 2 hours because of the down-time constraint. However, if the unit is needed, it can be off for only the first 2 hours and turn on for the next 2 hours to satisfy the reserve constraints. This will violate the downtime constraint. The pseudo code for the third rule is as follows:

Feasibility rule #3
If (φ = 0)
  While (F≠ Φ)
    a. Choose the unit ***i*** with the highest $\frac{\boldsymbol{p_i^{max}}}{\boldsymbol{c_i^p}}$ from **F**;
    b. Turn unit on earlier at time unit has been turned off by default (this time might be at previous period) and keep it on up to current time;
    c. Take unit ***i*** out from set **F**;
    d. **If** ($\sum_{i\in(onunits)} \mathrm{P}_i^{max} > D_{jt} + R_{jt}^{min}$)
      i. φ ← 1;
      ii. No more unit to be turned on
    Endif
  Endwhile
Endif

The last feasibility rule has to be applied if the first three rules do not take the lead to a feasible solution. In this rule, the units where their maintenance outage is started within the previous period without any power, and are on the maintenance outage at this time, are possible candidates to be turn on (back into operation).

If the feasibility rules failed to find a feasible solution up to the present step, the recent ants' group passes and the next group of ants will begin from the beginning to get a feasible solution. The pseudo code for the fourth rule is presented in the following:

Feasibility rule #4

If (φ = 0)
    While (O≠ Φ**)** /***set **O** belongs to previous period***/**
        a. Choose the unit ***i*** with lowest $\frac{c_i^p}{p_i^{max}\times(oph_i^{maxav}-pend_{i,j})}$;
        b. Revise unit maintenance outage at previous period and return unit ***i*** to the system;
        c. Turn it off up to current hour and turn it on now;
        d. Take unit ***i*** out from set **O**;
        e. **If** ($\sum_{i\in(onunits)} \mathrm{P}_i^{max} > D_{jt} + R_{jt}^{min}$)
            i. φ ← 1;
            ii. No more unit to be turned on
        Endif
    Endwhile
Endif

3.2.3. Calculating the amount of production for online units

The production for each online unit is calculated after a feasible solution is found for the GMS problem. The amount of production is to determine if this amount is equal to the load demand, thus the load balance constraint is fulfilled. The pseudo code for the calculation is as follows:

For each $\boldsymbol{j} \in \boldsymbol{J}$
  For each $\boldsymbol{t} \in \boldsymbol{T}$
    1. Set the production amount of online units to minimum;
    2. **While** $\left(D_{jt} > \sum_{i\in(onunits)} P_{ijt}\right)$
        i. Select online unit ***i*** that has the lowest production cost and has not been chosen before;
        ii. Set production of unit ***i*** to min $(p_i^{max} - p_i^{min}, D_{jt} - \sum_{i\in(onunits)} P_{ijt})$
      Endwhile
  Endfor
Endfor

4. Experimental design

Experiments have been performed to evaluate the performance of the multi-objective PACS algorithm. All experiments were conducted using Python 3.7 programming language on a machine with 16 GB RAM in a Windows 10 environment. The benchmark elements of the experiments' evaluation comprises the IEEE-RTS systems dataset with 32, and 36 units as used in [1, 4, 33]. Four metrics which comprise coverage, distance to Pareto front, overall Pareto spread, the number of obtained non-dominated solutions, and grey relational grade (GRG). These metrics are also used in [34] in their study on multi-objective scheduling problems. In particular, the GRG metric is used in [35, 36] in their multi-objective studies. Friedman test is also used to show the significant of the algorithm's performance.

Four (4) algorithms related to GMS problems which are commoly, and recently used in the literature within the domain of GMS are used as benchmark algorithms in the comparison. The algorithms are NSGAII, SPEA2, MOSA, and MOPSO [9, 18, 20, 23, 25, 26, 34, 37-39]. These argorithms are multi-objective metaheuristic search algorithms, which are classified as intelligent methods, and Pareto approach has been adopted. The NSGAII, and SPEA2 algorithms are classified as metheuristic and also evolutionary algorithms. The MOSA algorithm is metheuristic and classified as local search algorithms. Finally, MOPSO is classified under metheuristic and also swarm intelligence algorithms. The multi-objective algorithms can cater one problem with a multi-objective (i.e., the unit commitment problem implemented in conjunction with unit maintenance scheduling problem, with consideration the objective of cost minimization, reliability maximization, and violation minimization). The proposed multi-objective GMS model and sequential maintenance strategy based on operational hours have been implemented in all the algorithms

The parameter settings for the experiments are adopted from [1]. The parameters and the values for the algorithms are adopted from previous GMS studies. Different parameters are used in each study, such as parameters for PACS algorithm from [4]; for NSGAII and SPEA2 from [34]; for MOSA from [23]; and, finally for MOPSO from [37-39].

5. Results and discussion

This section presents the performance evaluation of the proposed multi-objective PACS algorithm. The results of the proposed algorithm are compared with the four benchmark algorithms based on several performance metrics related to the three (3) objective functions. The best results are highlighted.

Table 3 and 4 present the results for two (2) test systems based on three (3) objectives: cost, reliability (gross reserve), and convenience (violation). In general, the results from Table 3 and 4 show that the PACS algorithm outperforms other algorithms in terms of cost and reliability but is at par with MOSA for the violation objective. The Grey Relational Analysis method in [40] was adopted to select the best solution with low cost, high reliability, and low violation in Table 3 and 4.

Fig. 1 and 2 show the schedule for window [3000-5000] used with 32 & 36 unit system for one year (i.e. 52 periods). However it will take 53 periods if the unit enter for maintenance in period 52, the maintenance for that unit will exceed to period 53, because it considered the maintenance duration is identical for all units (i.e., two weeks).

Table 3. Results with 32-unit system

Objectives	**Maintenance window (hours)**	**PACS**	**NSGAII**	**SPEA2**	**MOSA**	**MOPSO**
Cost	[1000-2000] [1]	**205,146,072.03**	206,604,384.38	206,568,158.38	206,634,273.82	206,735,374.23
	[1000-3000]	**190,157,169.39**	202,247,526.91	202,593,302.37	202,107,932.49	202,633,621.08
	[2000-3000] [1]	**185,775,386.64**	192,031,766.33	191,007,725.92	192,345,982.42	190,955,430.31
	[2000-4000]	**178,487,720.69**	192,667,426.86	190,463,356.98	192,116,673.00	190,968,332.48
	[3000-4000] [1]	**179,335,380.59**	180,367,100.03	180,192,347.76	179,576,005.54	180,099,557.96
	[3000-5000]	**173,935,482.76**	181,265,811.21	179,798,368.61	181,116,547.97	179,844,689.91
Reliability	[1000-2000] [1]	**1,431,412.00**	1,424,785.00	1,425,841.00	1,419,417.00	1,415,644.00
	[1000-3000]	**1,454,207.00**	1,442,805.00	1,445,844.00	1,438,357.00	1,443,378.00
	[2000-3000] [1]	**1,464,603.00**	1,453,328.00	1,438,035.00	1,457,727.00	1,429,630.00
	[2000-4000]	1,450,120.00	**1,458,812.00**	1,450,718.00	1,452,353.00	1,452,287.00
	[3000-4000] [1]	**1,458,503.00**	1,455,497.00	1,458,059.00	1,443,849.00	1,452,430.00
	[3000-5000]	**1,476,805.00**	1,470,602.00	1,446,993.00	1,466,722.00	1,454,947.00
Violation	[1000-2000] [1]	**12**	14	13	**12**	13
	[1000-3000]	**0**	1	1	**0**	1
	[2000-3000] [1]	**0**	1	1	**0**	1
	[2000-4000]	**0**	1	1	**0**	1
	[3000-4000] [1]	**0**	1	1	**0**	1
	[3000-5000]	**0**	1	1	**0**	1

Table 4. Results with 36-unit system

Objectives	**Maintenance window (hours)**	**PACS**	**NSGAII**	**SPEA2**	**MOSA**	**MOPSO**
Cost	[1500-2500]	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible
	[1500-3000]	363,550,376.90	**359,775,789.69**	360,580,576.79	360,577,078.12	360,104,489.48
	[2000-3000]	362,531,811.91	360,057,115.47	**359,152,332.89**	359,272,269.41	360,394,194.66
	[2000-4000]	**345,982,571.97**	357,650,404.50	358,072,000.87	354,220,012.33	357,434,265.27
	[3000-4000]	**344,402,896.69**	345,577,570.97	345,984,272.55	345,693,900.02	345,182,624.24
	[3000-5000]	**336,980,517.06**	343,973,413.00	342,525,154.58	345,104,911.49	344,931,659.59
Reliability	[1500-2500]	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible
	[1500-3000]	2,397,544.00	**2,399,419.00**	2,388,078.00	2,384,118.00	2,385,822.00
	[2000-3000]	**2,404,129.00**	2,396,192.00	2,373,904.00	2,397,081.00	2,389,507.00
	[2000-4000]	2,409,526.00	2,399,400.00	**2,410,216.00**	2,400,994.00	2,402,316.00
	[3000-4000]	2,401,092.00	2,412,488.00	**2,413,045.00**	2,411,542.00	2,406,736.00
	[3000-5000]	**2,414,183.00**	2,394,430.00	2,399,111.00	2,411,148.00	2,413,521.00
violation	[1500-2500]	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible
	[1500-3000]	**0**	1	2	**0**	2
	[2000-3000]	**0**	1	1	**0**	1
	[2000-4000]	**0**	1	1	**0**	1
	[3000-4000]	**0**	1	1	**0**	1
	[3000-5000]	**0**	1	1	**0**	1

Received: March 28, 2022. Revised: June 15, 2022.

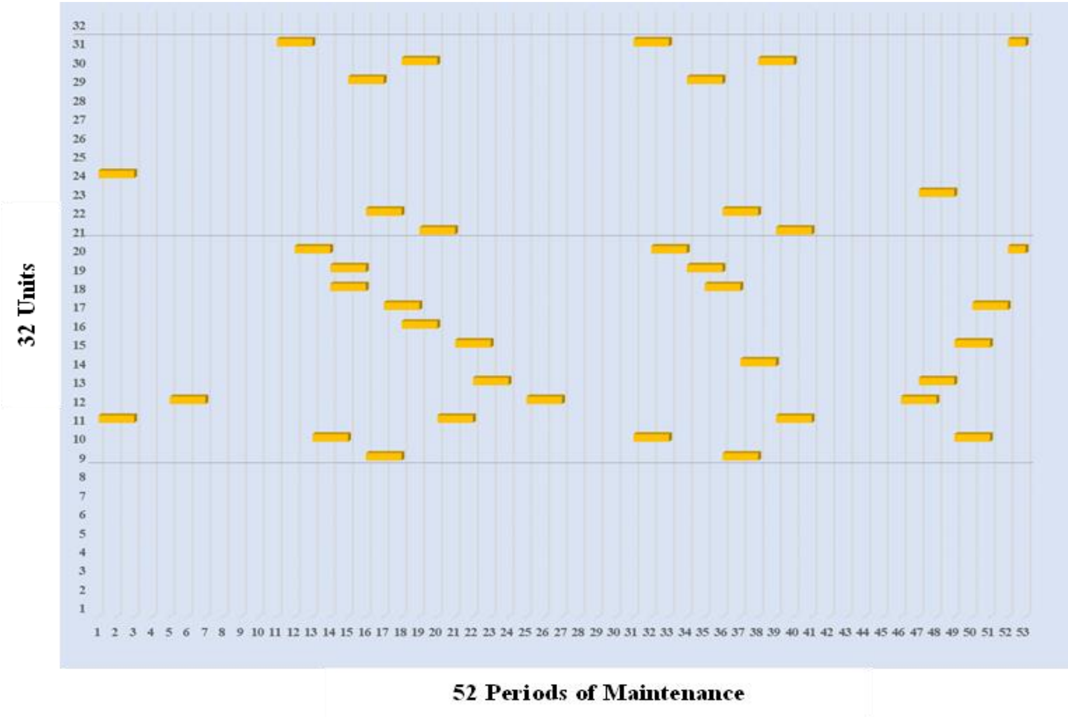

Figure. 1 Scheduling for window [3000-5000] with 32-unit system

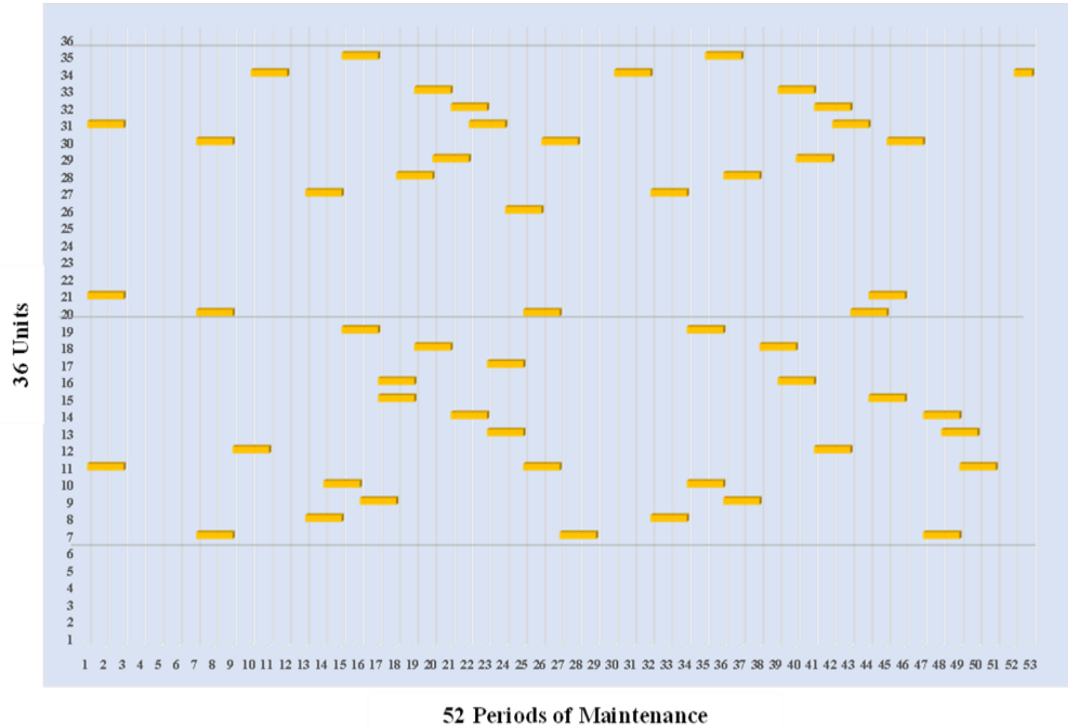

Figure. 2 Scheduling for window [3000-5000] with 36-unit system

Table 5 presents the GRG values for the PACS and benchmark algorithms. The GRG for each algorithm is calculated using the values of the three (3) objectives (i.e., operation cost, reliability, and violation). The best solution is indicated by the highest GRG value [40]. The PACS algorithm was able to obtain the best results for the 32-unit systems. For the 36-unit system, the PACS algorithm secured the second-best results at the early stages of the operation time, but outperformed other algorithms during other operation times. The GRG's improvement is also presented to show the performance improvement between PACS and every benchmark algorithm, as shown in Table 6. In general, PACS has better performance in all unit systems.

Table 5. Comparison for GRG and rank

Test Systems	Maintenance Window (hours)	PACS		NSGAII		SPEA2		MOSA		MOPSO	
		GRG	Rank	GRG	Rank	GRG	Rank	GRG	Rank	GRG	Rank
32-unit system	[1000-2000]	1	**1**	0.410	4	0.481	3	0.582	2	0.389	5
	[1000-3000]	1	**1**	0.361	5	0.385	3	0.559	2	0.363	4
	[2000-3000]	1	**1**	0.429	3	0.372	4	0.684	2	0.352	5
	[2000-4000]	0.778	**1**	0.556	3	0.352	5	0.581	2	0.365	4
	[3000-4000]	1	**1**	0.459	4	0.551	3	0.672	2	0.428	5
	[3000-5000]	1	**1**	0.458	3	0.350	5	0.645	2	0.374	4
36-unit system	[1500-2500]	Infeasible		Infeasible		Infeasible		Infeasible		Infeasible	
	[1500-3000]	0.712	2	0.833	**1**	0.479	5	0.678	3	0.515	4
	[2000-3000]	0.778	2	0.547	4	0.556	3	0.872	**1**	0.473	5
	[2000-4000]	0.962	**1**	0.336	5	0.556	3	0.598	2	0.362	4
	[3000-4000]	0.778	**1**	0.550	4	0.556	3	0.726	2	0.441	5
	[3000-5000]	1	**1**	0.345	5	0.384	4	0.699	2	0.536	3

Table 6. Comparison for GRG improvement

Test systems	Maintenance window (hours)	GRG Improvement (PACS, NSGAII)	GRG Improvement (PACS, SPEA2)	GRG Improvement (PACS, MOSA)	GRG Improvement (PACS, MOPSO)
32-unit system	[1000-2000]	**0.590**	**0.519**	**0.418**	**0.611**
	[1000-3000]	**0.639**	**0.615**	**0.441**	**0.637**
	[2000-3000]	**0.571**	**0.628**	**0.316**	**0.648**
	[2000-4000]	**0.222**	**0.426**	**0.196**	**0.413**
	[3000-4000]	**0.541**	**0.449**	**0.328**	**0.572**
	[3000-5000]	**0.542**	**0.650**	**0.355**	**0.626**
36-unit system	[1500-2500]	Infeasible	Infeasible	Infeasible	Infeasible
	[1500-3000]	-0.121	**0.233**	**0.034**	**0.197**
	[2000-3000]	**0.231**	**0.222**	-0.094	**0.305**
	[2000-4000]	**0.626**	**0.407**	**0.365**	**0.601**
	[3000-4000]	**0.228**	**0.222**	**0.051**	**0.337**
	[3000-5000]	**0.655**	**0.616**	**0.301**	**0.464**

To show the comparison statically, Table 7 summarizes the results obtained by Friedman test in which the *p* value is used to show the significance of the proposed algorithm. The GRG results for all the maintenance windows have been used to calculate the *p* values for the two (2) unit systems. In Friedman test, the significant level is set to 0.05. The mean rank indicates the difference in performance between the algorithms. The highest rank, which reflects the best algorithm, is assigned to the biggest value of the mean rank. It can be seen that the proposed PACS algorithm outperforms other algorithms. The computed *p* values for all the unit systems are less than 0.05 which shows that there is a significant difference in terms the GRG values between the proposed PACS algorithm and the other four (4) multi-objective algorithms. This implies that the PACS algorithm is significantly better than the other comparative algorithms. The GRG metric has been chosen for this kind of comparison because, in generator maintenance scheduling, getting the optimal maintenance scheduling with low cost, high reliability, and low violation is of utmost importance. The cost, reliability, and violation values are included in the GRG calculation.

In addition, four metrics were also used in the performance analysis of the multi-objective algorithms as presented in [34]. The performance metrics used for comparison comprise coverage (C), distance to Pareto front ($D1_R$), overall Pareto spread (OS), and the number of the obtained non-dominated solutions (NO.).

The C(A,B) value signifies the percentage of solutions in B that are dominated by at least one solution of A. It is essential to calculate C(B, A) and

Table 7. Results of friedman test

Test systems	Algorithms	Mean rank	Ranking
32-unit system	PACS	5.00	1
	NSGAII	2.33	3
	SPEA2	2.17	4
	MOSA	4.00	2
	MOPSO	1.50	5
	P value	**0.000**	
36-unit system	PACS	4.60	1
	NSGAII	2.20	4
	SPEA2	2.40	3
	MOSA	4.00	2
	MOPSO	1.80	5
	P value	**0.017**	

Table 8. Comparison based on C metric

Test systems	Maintenance window (hours)	*C* (PACS, NSGAII)	*C* (NSGAII, PACS)	*C* (PACS, SPEA2)	*C* (SPEA2, PACS)	*C* (PACS, MOSA)	*C* (MOSA, PACS)	*C* (PACS, MOPSO)	*C* (MOPSO, PACS)
32-unit system	[1000-2000]	**0.99**	0.02	**0.99**	0.02	**0.81**	0.07	**0.78**	0.00
	[1000-3000]	**1.00**	0.00	**1.00**	0.00	**0.97**	0.11	**0.99**	0.00
	[2000-3000]	**0.99**	0.00	**1.00**	0.00	**0.95**	0.17	**0.96**	0.00
	[2000-4000]	**1.00**	0.00	**1.00**	0.00	**1.00**	0.00	**1.00**	0.00
	[3000-4000]	**1.00**	0.00	**1.00**	0.00	**0.96**	0.14	**0.99**	0.00
	[3000-5000]	**1.00**	0.00	**1.00**	0.00	**1.00**	0.00	**1.00**	0.00
36-unit system	[1500-2500]	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible
	[1500-3000]	**0.89**	0.01	**0.90**	0.00	**0.59**	0.14	**0.59**	0.00
	[2000-3000]	**0.51**	0.03	**0.47**	0.01	0.19	**0.46**	**0.41**	0.01
	[2000-4000]	**1.00**	0.00	**0.99**	0.00	**0.95**	0.27	**0.97**	0.00
	[3000-4000]	**0.97**	0.00	**0.97**	0.00	**0.76**	0.49	**0.60**	0.00
	[3000-5000]	**0.95**	0.00	**0.93**	0.00	**0.58**	0.43	**0.47**	0.00

Table 9. Comparison based on $D1_R$ metric

Test systems	Maintenance window (hours)	$D1_R$ (PACS)	$D1_R$ (NSGAII)	$D1_R$ (PACS)	$D1_R$ (SPEA2)	$D1_R$ (PACS)	$D1_R$ (MOSA)	$D1_R$ (PACS)	$D1_R$ (MOPSO)
32-unit system	[1000-2000]	**0.01**	0.27	**0.00**	0.23	0.01	0.01	**0.00**	0.01
	[1000-3000]	**0.00**	0.21	**0.00**	0.26	0.04	0.04	**0.00**	0.01
	[2000-3000]	**0.00**	0.28	**0.00**	0.16	0.10	**0.03**	**0.00**	0.01
	[2000-4000]	**0.00**	4.58	**0.00**	4.22	**0.00**	0.61	**0.00**	0.21
	[3000-4000]	**0.00**	0.56	**0.00**	0.37	0.06	**0.05**	**0.00**	0.02
	[3000-5000]	**0.00**	3.04	**0.00**	3.02	**0.00**	0.47	**0.00**	0.17
36-unit system	[1500-2500]	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible
	[1500-3000]	**0.00**	0.05	**0.00**	0.04	0.05	**0.01**	**0.00**	0.01
	[2000-3000]	**0.00**	0.05	**0.00**	0.06	2.01	**0.10**	**0.00**	0.03
	[2000-4000]	**0.00**	0.51	**0.00**	0.53	0.16	0.16	**0.00**	0.03
	[3000-4000]	**0.00**	0.32	**0.00**	0.34	0.80	**0.14**	**0.00**	0.01
	[3000-5000]	**0.00**	0.96	**0.00**	0.92	0.44	**0.09**	**0.00**	0.02

A is better than B if C(A,B) > C(B,A) [34]. The metric of $D1_R$ is utilized to evaluate the distance of front A to a reference front (i.e., the Pareto-optimal front or a near Pareto-optimal front). The smaller the value of $D1_R$ (A) compared to $D1_R$ (B) the better the front A is [34]. The metric of overall Pareto spread (OS) measures the relative spread of two fronts. If OS(A, B)>1, front A is said to be favored to front B

Table 10. Comparison based on OS metric

Test systems	Maintenance window (hours)	*OS* (PACS, NSGAII)	*OS* (PACS, SPEA2)	*OS* (PACS, MOSA)	*OS* (PACS, MOPSO)
32-unit system	[1000-2000]	**1.60**	**2.11**	**5.19**	**6.72**
	[1000-3000]	**5.87**	**5.11**	**9.23**	**42.08**
	[2000-3000]	**3.07**	**3.35**	**5.38**	**10.54**
	[2000-4000]	**1.29**	**1.26**	**3.38**	**5.74**
	[3000-4000]	**1.84**	**2.22**	**4.80**	**7.30**
	[3000-5000]	**1.16**	**1.25**	**2.12**	**3.89**
36-unit system	[1500-2500]	Infeasible	Infeasible	Infeasible	Infeasible
	[1500-3000]	**4.25**	**5.44**	**9.41**	**22.46**
	[2000-3000]	**6.15**	**4.66**	**7.60**	**81.10**
	[2000-4000]	**3.50**	**3.36**	**7.82**	**19.95**
	[3000-4000]	**1.38**	**1.54**	**3.36**	**11.98**
	[3000-5000]	**2.40**	**2.54**	**10.03**	**17.76**

Table 11. Comparison based on NO. metric

Test systems	Maintenance window (hours)	*NO.* PACS	*NO.* NSGAII	*NO.* SPEA2	*NO.* MOSA	*NO.* MOPSO
32-unit system	[1000-2000]	**101.80**	99.71	99.64	10.20	8.80
	[1000-3000]	97.20	**100.00**	**100.00**	13.60	7.50
	[2000-3000]	98.90	**100.00**	**100.00**	16.50	7.20
	[2000-4000]	53.60	**100.00**	**100.00**	14.80	5.30
	[3000-4000]	59.90	**100.00**	**100.00**	18.50	9.10
	[3000-5000]	54.50	**100.00**	**100.00**	17.00	6.60
36-unit system	[1500-2500]	Infeasible	Infeasible	Infeasible	Infeasible	Infeasible
	[1500-3000]	**36.70**	8.70	7.90	4.20	3.70
	[2000-3000]	**28.80**	6.00	6.50	4.10	3.70
	[2000-4000]	43.00	**95.10**	94.70	12.50	5.90
	[3000-4000]	35.10	**64.20**	63.00	9.60	4.67
	[3000-5000]	30.70	**98.70**	98.20	9.40	4.10

with regard to the overall spread [34]. The last metric is the number of the obtained non-dominated solutions. Tables 8 to 11 display the comparison results of PACS with every other algorithm. In all the tables, the best results are highlighted.

In term of the coverage metric, the PACS algorithm outperformed the NSGAII, SPEA2, MOSA, MOPSO algorithms in all the maintenance windows with 32-unit system. With 36-unit system the PACS algorithm also outperformed other algorithms in all the maintenance windows, except for the window [2000-3000], where the MOSA outperforms the proposed algorithm.

For the distance to Pareto front metric, the proposed algorithm outperformed NSGAII, SPEA2, and MOPSO for the two-unit systems and in all the maintenance windows. However, in the comparison with MOSA, in 32-unit system there was mixed results, but in 36-unit system the MOSA outperforms the PACS algorithm. For the overall Pareto spread metric the proposed algorithm outperformed all the other four algorithms in all the maintenance windows for the two-unit systems. Finally, the number of obtained non-dominated solutions metric with 32-unit system the NSGAII, and SPEA2 algorithms outperformed other algorithm in four maintenance windows, while the proposed PACS algorithm was better in one maintenance window. With 36-unit system the NSGAII algorithm outperformed other algorithms with three maintenance windows, while the PACS algorithm was better in two maintenance windows.

It is observed that the infeasible case only appeared in the early stage of operation time for the 36-unit system. This result is the same as in [4]. In the window [1500-2500] many units entered for maintenance. Thus, the high load demand is not satisfied, and the infeasible case occurs. This is expected in the 36-unit system where the demand is double of the demand used with 32-unit system. It can be summarised that, in all the test unit systems, the PACS algorithm has the best performance in terms of GRG, coverage, distance to Pareto front and overall Pareto spread metrics while the NSGAII has the best result in terms of the number of obtained non-dominated solutions. The statical test shows that the performance of the PACS was significantly better than the four algorithms.

6. Conclusion

Performance evaluation of the proposed multi-objective PACS algorithm has shown that the proposed algorithm is superior to the benchmark algorithms in producing the solution for the GMS in terms of the multi-objective based on four metrics (i.e., GRG, coverage, distance to Pareto front, and overall Pareto spread), but with number of obtained non-dominated solutions metric the NSGAII algorithm was better. In addition, the Friedman test showed the significance performance of the proposed algorithm. As a result, from these comparisons, the proposed multi-objective PACS algorithm have proved it efficiency in terms of solution quality to reach the Pareto-optimal front or a near Pareto-optimal front in producing scheduling solution for generator maintenance. The PACS algorithm is a multi-objective metheuristic algorithm and swarm intelligence based algorithms. The proposed algorithm is based on Pareto approach, and developed to cater two problems with three objectives (i.e., the problems of GMS and unit commitment, in addition to the objectives of cost minimzation, reliablity maximization, and the violation minimzation). Single colony that comprises of several groups of ants has been used in the implementaion of the algorithm.

Future work related to method such as enhancement in the algorithm to increase its performance. The performance of the algorithm decreases as the data size increases. Future work can also look into the PACS algorithm working with big datasets by using multiple colonies instead of a single colony. Multiple colonies of ants have shown to be advantageous when the problem size gets bigger. In multiple colonies more than one colony is cooperating in order to better explore the problem space.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The draft has been prepared by the 1st author while the review and editing has been performed by the 2nd author.

References

[1] S. A. Muthana and K. R. K. Mahamud, “Generator Maintenance Scheduling Models for Electrical Power Systems : A Review”, *Int. J. Electr. Electron. Eng. Telecommun.*, Vol. 10, No. 5, pp. 1-12, 2021.

[2] G. Kamel, M. F. Aly, A. Mohib, and I. H. Afefy, “Optimization of a multilevel integrated preventive maintenance scheduling mathematical model using genetic algorithm”, *Int. J. Manag. Sci. Eng. Manag.*, Vol. 15, No. 4, pp. 247-257, 2020.

[3] V. Aristidis, “Meta-heuristic optimization techniques in power systems”, In: *Proc. of the 2nd IASME / WSEAS International Conference on Energy & Environment*, pp. 163-174, 2007.

[4] M. Fattahi, M. Mahootchi, H. Mosadegh, and F. Fallahi, “A new approach for maintenance scheduling of generating units in electrical power systems based on their operational hours”, *Comput. Oper. Res.*, Vol. 50, No. Feb 1, pp. 61-79, 2014.

[5] W. K. Foong, H. R. Maier, and A. R. Simpson, “Ant colony optimization for power plant maintenance scheduling optimization”, In: *Proc. of the 7th Annual Conference on Genetic and Evolutionary Computation*, pp. 249-256, 2005.

[6] M. Saffarian, S. M. Kazemi, and M. Baharshahi, “Generator maintenance scheduling using discrete firefly algorithm”, *Int. J. Nonlinear Anal. Appl.*, Vol. 11, No. 1, pp. 367-379, 2020.

[7] A. Vlachos, “Rank-based ant colony algorithm for a thermal generator maintenance scheduling problem”, *WSEAS Trans. Circuits Syst.*, Vol. 12, No. 9, pp. 273-285, 2013.

[8] Y. Xu, X. Han, M. Yang, M. Wang, X. Zhu, and Y. Zhang, “Condition-based midterm maintenance scheduling with rescheduling strategy”, *Int. J. Electr. Power Energy Syst.*, Vol. 118, No. Jun 1, p. 105796, 2020.

[9] B. G. Lindner, R. Brits, J. H. V. Vuuren, and J. Bekker, “Tradeoffs between levelling the reserve margin and minimising production cost in generator maintenance scheduling for regulated power systems”, *Int. J. Electr. Power Energy Syst.*, Vol. 101, No. Oct 1, pp. 458-471, 2018.

[10] J. Zhan, C. Guo, Q. Wu, L. Zhang, and H. Fu, “Generation maintenance scheduling based on multiple objectives and their relationship analysis”, *J. Zhejiang Univ. Sci. C*, Vol. 15, No. 11, pp. 1035-1047, 2014.

[11] G. Balaji, R. Balamurugan, and L. Lakshminarasimman, “Fuzzy clustered multi

objective differential evolution for thermal generator maintenance scheduling", *Int. J. Intell. Eng. Syst.*, Vol. 9, No. 1, pp. 1-13, 2016, doi: 10.22266/ijies2016.0331.01.

[12] A. Afshar, A. Kaveh, and O. R. Shoghli, "Multi-objective optimization of time-cost-quality using multi-colony ant algorithm", *Asian J. Civ. Eng.*, Vol. 8, No. 2, pp. 113-124, 2007.

[13] B. G. Lindner, "Bi-objective generator maintenance scheduling for a national power utility", *Doctoral Dissertation, Stellenbosch: Stellenbosch University*, 2017.

[14] S. Bechikh, R. Datta, and A. Gupta, "Recent advances in evolutionary multi-objective optimization", *Springer International Publishing Switzerland*, Vol. 20, 2016.

[15] A. T. D. Almeida, R. J. P. Ferreira, and C. A. V. Cavalcante, "A review of the use of multicriteria and multi-objective models in maintenance and reliability", *IMA J. Manag. Math.*, Vol. 26, No. 3, pp. 249-271, 2015.

[16] P. Ngatchou, A. Zarei, and M. A. E. Sharkawi, "Pareto multi objective optimization", In: *Proc. of the 13th International Conference on, Intelligent Systems Application to Power Systems*, pp. 84-91, 2005.

[17] D. Constantinou, "Ant colony optimisation algorithms for solving multi-objective power-aware metrics for mobile ad hoc networks", *Doctoral Dissertation, University of Pretoria*, 2010.

[18] S. Zhong, A. A. Pantelous, M. Goh, and J. Zhou, "A reliability-and-cost-based fuzzy approach to optimize preventive maintenance scheduling for offshore wind farms", *Mech. Syst. Signal Process.*, Vol. 124, No. 1 Jun, pp. 643-663, 2019.

[19] D. Zhang, W. Li, and X. Xiong, "Bidding based generator maintenance scheduling with triple-objective optimization", *Electr. Power Syst. Res.*, Vol. 93, No. Dec 1, pp. 127-134, 2012.

[20] X. D. Chen, J. P. Zhan, Q. H. Wu, and C. X. Guo, "Multi-objective optimization of generation maintenance scheduling", In: *Prof. of IEEE PES General Meeting| Conference & Exposition*, pp. 1-5, 2014.

[21] F. Yang and C. S. Chang, "Optimisation of maintenance schedules and extents for composite power systems using multi-objective evolutionary algorithm", *IET Gener. Transm. Distrib.*, Vol. 3, No. 10, pp. 930-940, 2009.

[22] M. A. Latify, H. Seif, and H. R. Mashhadi, "A strength Pareto evolutionary algorithm–based conflict assessment framework of electricity market participants' objectives in generation maintenance scheduling", *Eur. Trans. Electr. Power*, Vol. 23, No. 3, pp. 342-363, 2011.

[23] K. S. Moghaddam and J. S. Usher, "Maintenance scheduling of multi-component systems using multi-objective simulated annealing", In: *Proc. of the 2009 Industrial Engineering Research Conference*, pp. 2189-2194, 2009.

[24] M. Alrifaey, T. S. Hong, A. As'arry, E. E. Supeni, and C. K. Ang, "Optimization and selection of maintenance policies in an electrical gas turbine generator based on the hybrid reliability-centered maintenance (RCM) model", *Processes*, Vol. 8, No. 6, pp. 1-26, 2020.

[25] H. Guo, S. He, C. Huang, Q. Zhou, and Y. Zhao, "Generation maintenance scheduling based on multi-objective particle swarm optimization considering carbon emissions", In: *Proc. of 2021 IEEE 2nd China International Youth Conference on Electrical Engineering*, pp. 1-7, 2021.

[26] H. Abdollahzadeh, K. Atashgar, and M. Abbasi, "Multi-objective opportunistic maintenance optimization of a wind farm considering limited number of maintenance groups", *Renew. Energy*, Vol. 88, No. 1 April, pp. 247-261, 2016.

[27] M. Sharafi and T. Y. Elmekkawy, "Multi-objective optimal design of hybrid renewable energy systems using PSO-simulation based approach", *Renew. Energy*, Vol. 68, No. Aug 1, pp. 67-79, 2014.

[28] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey", *ACM Comput. Surv.*, Vol. 48, No. 1, pp. 1-35, 2015.

[29] C. G. Martínez, O. Cordón, and F. Herrera, "A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP", *Eur. J. Oper. Res.*, Vol. 180, No. 1, pp. 116-148, 2007.

[30] Y. Su and R. Chi, "Multi-objective particle swarm-differential evolution algorithm", *Neural Comput. Appl.*, Vol. 28, No. 2, pp. 407-418, 2017.

[31] M. J. Polenick and M. W. Scott, "Multiple gas turbine engines to normalize maintenance intervals", 2000, https://patentimages.storage.googleapis.com/f7/05/d8/a6bd196d7b9523/US6158207.pdf.

[32] Y. Zhao, V. Volovoi, M. Waters, and D. Mavris, "A sequential approach for gas turbine power plant preventative maintenance scheduling", *J.*

Eng. Gas Turbines Power, Vol. 128, pp. 796-805, 2006.

[33] M. Shahidehpour and M. Marwali, *Maintenance scheduling in Restructured Power Systems*, 2000.

[34] A. Berrichi, F. Yalaoui, L. Amodeo, and M. Mezghiche, "Bi-objective ant colony optimization approach to optimize production and maintenance scheduling", *Comput. Oper. Res.*, Vol. 37, No. 9, pp. 1584-1596, 2010.

[35] S. Jozić, D. Bajić, and L. Celent, "Application of compressed cold air cooling: Achieving multiple performance characteristics in end milling process", *J. Clean. Prod.*, Vol. 100, No. Aug 1, pp. 325-332, 2015.

[36] A. Panda, A. K. Sahoo, and A. K. Rout, "Multi-attribute decision making parametric optimization and modeling in hard turning using ceramic insert through grey relational analysis: A case study", *Decis. Sci. Lett.*, Vol. 5, No. 4, pp. 581-592, 2016.

[37] Y. Yare and G. K. Venayagamoorthy, "Optimal scheduling of generator maintenance using modified discrete particle swarm optimization", In: *Proc. of 2007 iREP Symposium-Bulk Power System Dynamics and Control-VII. Revitalizing Operational Reliability*, pp. 1-8, 2007.

[38] Y. Yare and G. K. Venayagamoorthy, "Comparison of DE and PSO for generator maintenance scheduling", In: *Proc. of 2008 IEEE Swarm Intelligence Symposium*, pp. 1-8, 2008.

[39] Y. Yare, G. K. Venayagamoorthy, and U. O. Aliyu, "Optimal generator maintenance scheduling using a modified discrete PSO", *IET Gener. Transm. Distrib.*, Vol. 2, No. 6, pp. 834-846, 2008.

[40] F. Kolahan and M. A. Moghaddam, "The use of Taguchi method with grey relational analysis to optimize the EDM process parameters with multiple quality characteristics", *Sci. Iran.*, Vol. 22, No. 2, pp. 530-538, 2015.