Engineering Optimization

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/geno20

Solution of optimal reactive power dispatch problem using pathfinder algorithm

Hamza Yapici

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2020.1839443

Solution of optimal reactive power dispatch problem using pathfinder algorithm

Hamza Yapici

Ereğli Vocational School, Necmettin Erbakan University, Konya, Turkey

ABSTRACT

The optimal reactive power dispatch (ORPD) problem, as a subproblem of optimal power flow, has significant effects in providing reliability and economic operation. In this article, a modified version of the pathfinder algorithm (PFA), which is inspired by the collective movement of a swarm led by one member, is proposed for solving the ORPD problem. The objective of this study is to minimize the power losses by adjusting the control variables. Numerical analyses are performed on 57-bus and 118-bus power systems. To show the performance and effectiveness of the modified pathfinder algorithm (mPFA), some well-known methods are used for comparison. Statistical tests are performed to assess the consistency and ranking of the proposed method. The results show that the mPFA achieved competitive results and obtained a competitive ranking with statistical analyses. The simulations show that the proposed method could be a superior algorithm for solving the ORPD problem.

ARTICLE HISTORY
Received 15 January 2020
Accepted 14 October 2020

KEYWORDS
Minimization of power loss; optimal reactive power dispatch; pathfinder algorithm; statistical tests

1. Introduction

The optimal reactive power dispatch (ORPD) problem has a key role in the operation and control of power systems owing to its significant effects on reliability and economical management. In this problem, settings of control variables are adjusted to optimize a certain object, taking into account the equality and inequality constraints. Therefore, the ORPD problem, a subproblem of optimal power flow, is modelled using mixed-integer nonlinear programming (Khazali and Kalantar 2011).

The difficulties of adjusting the parameters for electric power networks have led researchers to propose new optimization approaches as well as the constraint handling methods. Optimization methods are used to define the optimal value of control variables, while constraint handling methods assist these methods to overcome the limitations. Besides mathematical approaches, heuristic algorithms for optimization are attractive to researchers in terms of solving the ORPD problem. In the literature, many techniques have been proposed to solve the ORPD problem, such as evolutionary and heuristic-based techniques, resulting in high-quality solutions. Some of these methods can be listed as follows: genetic algorithm (GA) (Durairaj, Devaraj, and Kannan 2006), improved genetic algorithm (IGA) (Devaraj and Roselyn 2010), evolutionary programming (EP) (Wu and Ma 1995), differential evolution (DE) (Varadarajan and Swarup 2008), modified teaching–learning algorithm with differential evolution (MTLA-DE) (Ghasemi *et al.* 2014), gravitational search algorithm (GSA) (Duman *et al.* 2012), big bang–big crunch (BB-BC) (Zandi, Afjei, and Sedighizadeh 2012), particle swarm optimization (PSO) (Zhao, Guo, and Cao 2005), seeker optimization algorithm (SOA) (Dai *et al.* 2009),

CONTACT Hamza Yapici hyapici@erbakan.edu.tr

Supplemental data for this article can be accessed here. https://doi.org/10.1080/0305215X.2020.1839443

artificial bee colony (ABC) (Le Dinh, Vo Ngoc, and Vasant 2013), comprehensive learning particle swarm optimization (CLPSO) (Mahadevan and Kannan 2010), particle swarm optimization with an ageing leader and challengers (ALC-PSO) (Singh, Mukherjee, and Ghoshal 2015), chaotic krill herd algorithm (CKHA) (Mukherjee and Mukherjee 2016), grey wolf optimizer (GWO) (Sulaiman *et al.* 2015), moth–flame optimizer (MFO) (Mei *et al.* 2017), ant lion optimizer (ALO) (Mouassa, Bouktir, and Salhi 2017), backtracking search (BS) method (Shaheen, El-Sehiemy, and Farrag 2018), whale optimization algorithm (WOA) (Medani, Sayah, and Bekrar 2018), Gaussian bare-bones water cycle optimizer (GBBWCO) (Heidari, Abbaspour, and Jordehi 2017), Jaya optimization algorithm (JAYA) (Barakat *et al.* 2018; Das *et al.* 2020), improved social spider algorithm (ISSA) (Nguyen and Vo 2019), improved ant lion optimizer (IALO) (Li *et al.* 2019), modified version of sine–cosine method (ISCA) (Abdel-Fatah, Ebeed, and Kamel 2019), success history-based adaptive differential evolution (SHADE) (Biswas *et al.* 2019), tree seed algorithm (TSA) (Üney and Cetinkaya 2019), enhanced Jaya optimization method (e-JAYA) (Barakat *et al.* 2019), modified stochastic fractal search algorithm (MSFSA) (Nguyen *et al.* 2019) and a modified version of differential evolution (DEa-AR) (Awad *et al.* 2019). Also, some different approaches are semi-definite programming (SDP) (Davoodi *et al.* 2019) and tight-and-cheap conic relaxation approach (TCCR) (Bingane, Anjos, and Le Digabel 2019).

Evolutionary-based approaches have key abilities to obtain global optima and handle non-convex as well as discontinuous objectives. However, these approaches have disadvantages in solving the ORPD problem. They are insufficient in analysing discrete or integer problems and may not be able to find the optimal solution in finite time. Furthermore, the swarm intelligence (SI)-based methods have fewer operators to be adjusted and adapt easily with minimum revision for different areas. But they are highly redundant and it is difficult to enforce control over the herds. Their complex systems cause unforeseeable results and take time owing to their rich hierarchies (Ayan and Kılıc 2012; Saddique *et al.* 2020).

Among these approaches, several recently introduced methods can be explained as follows. A modified version of PSO handling a pseudo-gradient search method (PSO-IPGS) was proposed by Polprasert, Ongsakul, and Dieu (2016). The simulations have been performed on 30-bus and 118-bus test systems. Using this approach, PSO determines an effective direction. PSO-IPGS outperforms all other versions. However, PSO and its modified versions cannot generate high-quality solutions using low population size or iteration numbers. In terms of low population size, they can become stagnated. Moreover, they are highly dependent on the initial parameters. Nguyen *et al.* (2019) proposed a modified stochastic search algorithm (MSFS), making three changes in three new solution generations of the original version of this algorithm. In the first generation, only one equation is used and the other one is eliminated; however, in the second and third generations, a new method for generating new solutions is introduced. Using the new technique, the proposed method shows better performance than the original version. Furthermore, old solutions are modified to be updated. The proposed method was tested on 30-bus and 118-bus test systems. Although these additional modifications provide a shorter computational time than the original version, the results have not given sufficient evidence for comparing the computation time with other methods. The proposed method produced high-quality solutions in the early stages of the process, but it was not able to effectively generate new solutions over the course of iterations. Li *et al.* (2019) proposed the IALO to solve the ORPD problem. The authors introduced two improvements. The first improvement is to use a new technique instead of roulette wheel selection. This improvement separates all solutions into potential and non-potential groups. A new formula is applied with the second improvement to speed up the process, reduce the number of computation steps and decrease implementation time. This method was tested on 30-bus, 57-bus and 118-bus test systems. IALO outperformed all other methods in comparison of computational time. However, these modifications may cause the individual to move towards any non-promising area.

In addition, different constraint handling methods have been used, such as the quadratic penalty function, adaptive penalty function and alternative penalty function, to handle all equality and inequality constraints. But adaptive and alternative constraint handling methods contain fewer variables. This causes solutions with low quality to be generated. Also, the quadratic penalty function is time consuming in finding the optimal values and significantly affects the solution in large-scale problems.

This article proposes a new method, named the modified pathfinder algorithm (mPFA), to solve the ORPD by optimizing the adjustable control variables. The pathfinder algorithm (PFA) is a novel metaheuristic method which was originally developed by Yapici and Cetinkaya (2019). The PFA has been applied to several research areas in terms of its ability to make the transition between exploration and exploitation, avoiding local optima and achieving a better convergence rate. The PFA has been used for network reconfiguration problems (Nguyen *et al.* 2020), and automatic generation control with tilt–integral-derivative (TID) and proportional–integral-derivative (PID) (Priyadarshani, Subhashini, and Satapathy 2020). However, the PFA has a key disadvantage in that its searching ability decreases in extremely high dimensions. In an effort to improve the performance of the PFA, several versions have been introduced, such as modified PFA with DE, in view of convergence speed and avoidance of local optima (Qi, Yuan, and Song 2020), and improved PFA using a quasi-oppositional learning mechanism to improve the convergence speed and chaos theory to improve the exploration part (Yuan, Li, and Yousefi 2021). The particles of the PFA are led by a leader member, which is completely different from the follower members in terms of its mathematical model. This difference facilitates the transition between exploration and exploitation. The followers move towards the next position in a non-regular order, and thus, they can explore the search space effectively. Therefore, the PFA has a robust capability to solve optimization problems and can find better solutions. The PFA is introduced as a simple method, easily adapting to optimization problems as it has several parameters that can be adjusted, and is more effective in converging to the global optimum. Because of the reduced performance of the PFA in high dimensions, several modifications for parameters have been proposed to generate better solutions. The intention here is to modify the original PFA without making too many changes to the general mechanism.

The simulations were performed on the Institute of Electrical and Electronics Engineers (IEEE) 57-bus and 118-bus test systems. The objective is to minimize the power loss of these power systems. The quadratic penalty function was used for handling the constraints. Owing to the characteristics of the mPFA, it can overcome the disadvantages explained above and achieve competitive results in numerical analysis. Moreover, some statistical tests were carried out to rank the proposed method and assess its consistency: the Kolmogorov–Smirnov test, Friedman test and Wilcoxon test (Pesaran and Timmermann 1992; Simard and L'Ecuyer 2011; Zimmerman and Zumbo 1993). To the author's knowledge, the Kolmogorov–Smirnov test is used for the first time in this area. The outstanding contributions of this study can be summarized as follows:

(1) To minimize power loss of power systems in the context of providing reliability and economic operation, the modified version of the PFA is proposed for the first time in solving the ORPD problem.
(2) To overcome premature convergence and the stagnation in local optima to achieve better performance with high-quality solutions, the parameters of the PFA are modified without changing the general mechanism.
(3) The conventional quadratic penalty handling method is considered for an effective and competitive comparison by including bus voltage, reactive power of generators and line constraints, although it is time consuming.
(4) According to the numerical results on different IEEE power systems, the mPFA could obtain better results than the PFA and other methods in the context of reducing power losses. It could also outperform other recently introduced methods in terms of convergence and statistical results.

The remainder of this article is arranged as follows. The formulation of ORPD is explained in Section 2; a short explanation of the PFA and its modification is then presented in Section 3 and the implementation of the proposed method is explained in Section 4. Sections 5 and 6 comprise the results and discussions of the study. Finally, Section 7 includes the conclusions.

2. Formulation of the ORPD problem

The ORPD problem has an important role in operating and controlling power systems. It includes some parameters to be adjusted, such as the tap settings of transformers, shunt capacitors and voltage magnitude, named the control parameters. The tap settings of transformers and shunt capacitors are discrete or integer values. However, voltage magnitude is assumed as a continuous variable (Dai *et al.* 2009; Yapıcı and Çetinkaya 2017). The ORPD problem, known as a complex mixed-integer nonlinear optimization problem, can be defined as follows:

$$\begin{array}{ll} \text{Minimize} & f(\vec{x},\vec{X}) \\ \text{Subject to} & G(\vec{x},\vec{X}) = 0 \\ & H(\vec{x},\vec{X}) \geq 0 \end{array} \tag{1}$$

where $\vec{x}$ is the vector of control variables, $\vec{X}$ is the vector of dependent variables, and f is the objective function. Here, $\vec{x} = [V_G T Q_C]^T$ and $\vec{X} = [V_L Q_G S]^T$. Also, G is the equality constraint and H is the inequality constraint. The equality constraints consist of the balanced equations of active and reactive powers, whereas inequality constraints include the voltage of generators, tap setting of transformers and capacity of compensators (Mei *et al.* 2017). The objective function was solved subject to various equality and inequality constraints in the ORPD problem. The formulations of equality and inequality constraints are given in Equations (2) and (3), respectively. The equality constraints formulated in Equation (2) are the active and reactive power balance equations. By satisfying these conditions, the voltage and the frequency of buses in a power system can be adjusted. Thus, stable working can be guaranteed for a power system. The first formula is the active power balance equation and the second one is the reactive power balance equation. The constraints imposed in Equation (3) are inequalities used for the limitation of reactive power sources, tap setting of transformers, active power generations, reactive power generations, bus voltages and line flows (Yapıcı and Çetinkaya 2017).

$$\begin{aligned} P_{Gi} - P_{Di} &= V_i \sum_{j \in N_i} V_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}), i \epsilon N_O \\ Q_{Gi} - Q_{Di} &= V_i \sum_{j \in N_i} V_j (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}), i \epsilon N_Q \end{aligned} \tag{2}$$

$$\begin{array}{ll} Q_{Ci}^{min} \leq Q_{Ci} \leq Q_{Ci}^{max}, & i\epsilon N_C \\ T_i^{min} \leq T_i \leq T_i^{max}, & i\epsilon N_T \\ V_{Gi}^{min} \leq V_{Gi} \leq V_{Gi}^{max}, & i\epsilon N_G \\ P_{Gi}^{min} \leq P_{Gi} \leq P_{Gi}^{max}, & i\epsilon N_G \\ Q_{Gi}^{min} \leq Q_{Gi} \leq Q_{Gi}^{max}, & i\epsilon N_G \\ V_{Li}^{min} \leq V_{Li} \leq V_{Li}^{max}, & i\epsilon N_B \\ S_{Ki} \leq S_{Ki}^{max}, & i\epsilon N_K \end{array} \tag{3}$$

where P_{Gi} is the active power generated, P_{Di} is the power demanded, N_i is the number of neighbouring buses, N_O is the number of total buses not including the slack bus, N_Q is the number of PQ buses, Q_{Gi} is the reactive power generated, Q_{Di} is the reactive power demanded, G_{ij} and B_{ij} are the conductance and susceptance between buses i and j, respectively, Q_{Ci} is the value of shunt capacitor i, T_i is the tap setting of transformer i, V_{Gi} is the voltage of generator i, V_{Li} is the voltage of load bus i, S_{Ki} is the load flow in branch i, N_C is the number of shunt capacitors, N_T is the number of transformers, N_G is the number of generators, N_B is the number of buses, and N_K is the number of branches.

The control variables are self-constrained. However, dependent variables such as reactive power injected by PV buses, voltage magnitudes of PQ buses and power flow limits are added as penalty terms to the objective function. It is important to mention here that the active power outputs of generators are determined before the process and dedicated as input data. Finally, the objective function and fitness function can be expressed as the formulae given in Equations (4) and (5), respectively (Li *et al.* 2019).

$$f = \min P_{loss} = \sum_{i=1}^{N_B} \sum_{j=1, j\neq i}^{N_B} g_{ij}(V_i^2 + V_j^2 - 2V_iV_j\cos\theta_{ij}) \tag{4}$$

$$f' = f + \mu_v \sum_{N_v^{lim}} \Delta V_L^2 + \mu_q \sum_{N_q^{lim}} \Delta Q_G^2 + \mu_s \sum_{N_s^{lim}} \Delta S_i^2 \tag{5}$$

where f' is the fitness function, P_{loss} is total power loss, i is the index of buses, j is the index of buses neighbouring bus i, g_{ij} is the conductance between buses i and j, V_i is the voltage of bus i, V_j is the voltage of bus j, θ_{ij} is the angle between buses i and j, μ_v, μ_q and μ_s are constants and are chosen as 1000, and N_v^{lim}, N_q^{lim} and N_s^{lim} are the number of load buses, generators and branches taken out of limits, respectively. If $V_L < V_L^{min}$, then $\Delta V_L = V_L^{min} - V_L$, and if $V_L > V_L^{max}$, then $\Delta V_L = V_L - V_L^{max}$. Therefore, if $Q_G < Q_G^{min}$, then $\Delta Q_G = Q_G^{min} - Q_G$, and if $Q_G > Q_G^{max}$, then $\Delta Q_G = Q_G - Q_G^{max}$. Finally, if $S_i > S_i^{max}$, then $S_i = S_i - S_i^{max}$. Here, 'min' and 'max' are the minimum and maximum values, respectively. Furthermore, when the dependent variables obtained after power flow exceed their limits, they are added to the objective function as penalty terms. Otherwise, the penalty terms would be equal to zero.

3. Pathfinder algorithm (PFA)

The PFA is an SI-based method inspired by the collective movement of swarms with a leader member. This method allows all members of swarm to explore the search space randomly, while they decide to move towards any location by following the leader. When a member locates in the most promising area, then this individual is chosen as the leader. In particular, it is worth stating that the movements of the leader and the members are completely different mathematically. The leader member is called the pathfinder. It saves the best solution in each iteration. The other members use Equation (6), whereas the pathfinder moves towards the next location using Equation (7).

$$x_i^{k+1} = x_i^k + R_1(x_j^k - x_i^k) + R_2(x_p^k - x_i^k) + \varepsilon, i\epsilon[2, N_{pop}] \tag{6}$$

$$x_p^{k+1} = x_p^k + 2r_3(x_p^k - x_p^{k-1}) + A \tag{7}$$

where k is the current iteration, x_i is the position vector of member i, x_j is the position vector of member j, R_1 and R_2 are random variables which are equal to αr_1 and βr_2, and N_{pop} is the population size. Here, r_1, r_2 and r_3 are random variables generated in the range of [0,1] uniformly, x_p is the position vector of the pathfinder, and ε and A are the vibration and fluctuation coefficients, respectively. ε and A are generated over the course of iterations via Equations (8) and (9), respectively. Also, α and β are selected randomly in the range of [1,2] in each iteration.

$$\varepsilon = \left(1 - \frac{k}{k_{max}}\right) u_1 D_{ij}, D_{ij} = ||x_i - x_j|| \tag{8}$$

$$A = u_2 e^{-\frac{2k}{k_{max}}} \tag{9}$$

where u_1 and u_2 are random variables in the range [−1,1], and D_{ij} is the distance between two members.

3.1. Modified pathfinder algorithm (mPFA)

In this study, some modifications have to be made. In contrast to the benchmark problems considered for continuous space, the bounds of some variables in the ORPD problem are constricted. Therefore, ε and A have been modified for an efficient search. Because of these modifications, they change within small ranges. Thus, any member can move towards the next position with low vibration around the promising area. These are modified as follows:

$$\varepsilon = 0.1\varepsilon \tag{10}$$

$$A = 0.001A \tag{11}$$

A and ε have key abilities to support the random walk as well as making the transition between exploration and exploitation. Thus, A and ε, to maintain the random walk, should have proper values. For the ORPD problem, the main modification is to select A and ε close to zero to provide the exploration in the initial steps of the iterations and then the exploitation at the end of the process. A suitable value for A around zero allows the pathfinder to move towards the next location with small steps. When it is selected as zero, then the random walk does not occur. Therefore, A is multiplied by 0.001. Moreover, during the many executions, it was observed that when $\varepsilon \ll 1$, then follower members can change their location with small steps. However, adjusting ϵ around 0 may not allow the followers to find the most promising solution effectively. So, ε is multiplied by 0.1.

In addition, there are different end criteria in the literature, including a fixed number of iterations, an unchanging string with a certain value, and no change in average fitness value after a few iterations (Yapıcı and Çetinkaya 2017). In this article, the maximum iteration number is used as the stop criterion. The flowchart of the mPFA is shown in Figure 1. The pseudo-code of the mPFA is given in Figure S1 in the Supplementary Material.

4. Implementation of the mPFA

In this section, the implementation of the mPFA for the ORPD problem is introduced for power loss minimization. The simulations are also performed on IEEE 57-bus and IEEE 118-bus test systems. The position of members consists of control variables such as generator voltages, tap settings and values of shunt compensators, where $x_i = \{V_{G1,i}, \ldots, V_{GN_G,i}, T_{1,i}, \ldots, T_{N_T,i}, Q_{C1,i}, \ldots, Q_{CN_C,i}\}^T$, $i = 1, \ldots, N_{pop}$. The positions of members must be within the lower and upper limits. The lower and upper bounds are constructed as $\{V_{G1,min}, \ldots, V_{GN_G,min}, T_{1,min}, \ldots, T_{N_T,min}, Q_{C1,min}, \ldots, Q_{CN_C,min}\}^T$ and $\{V_{G1,max}, \ldots, V_{GN_G,max}, T_{1,max}, \ldots, T_{N_T,max}, Q_{C1,max}, \ldots, Q_{CN_C,max}\}^T$, respectively. Then, the variables are compared to the lower and upper bounds after generating new populations. When x_i violates its lower bound, it is fixed to the lower bound. When x_i violates its upper bound, it is fixed to the upper bound. MATPOWER software (Wang *et al.* 2007; Zimmerman, Murillo-Sánchez, and Thomas 2010) is used to calculate Newton-based power flow for the ORPD problem. It should be noted that the ORPD problem is a real-world problem. Although variables are continuous in mathematical optimization problems, some variables can be discrete or integer in real-world problems. To use these variables, the proposed method explores the search space continuously and then evaluates the fitness function using an interruption to cut the matching dimensions of members into integers. In the ORPD problem, the tap position and reactive power source installation are discrete variables while the generator voltage can be assumed to be a continuous variable. Moreover, inequality constraints for the dependent variable are handled as follows: when a variable violates its limits, it is fixed to its upper or lower limit. The implementation of the algorithm is clarified in Figure 2.

From the explanation given in Figure 2, each position vector is addressed to power flow data, and the position vectors are then executed in MATPOWER software. All members are updated using '*if then*' rules.

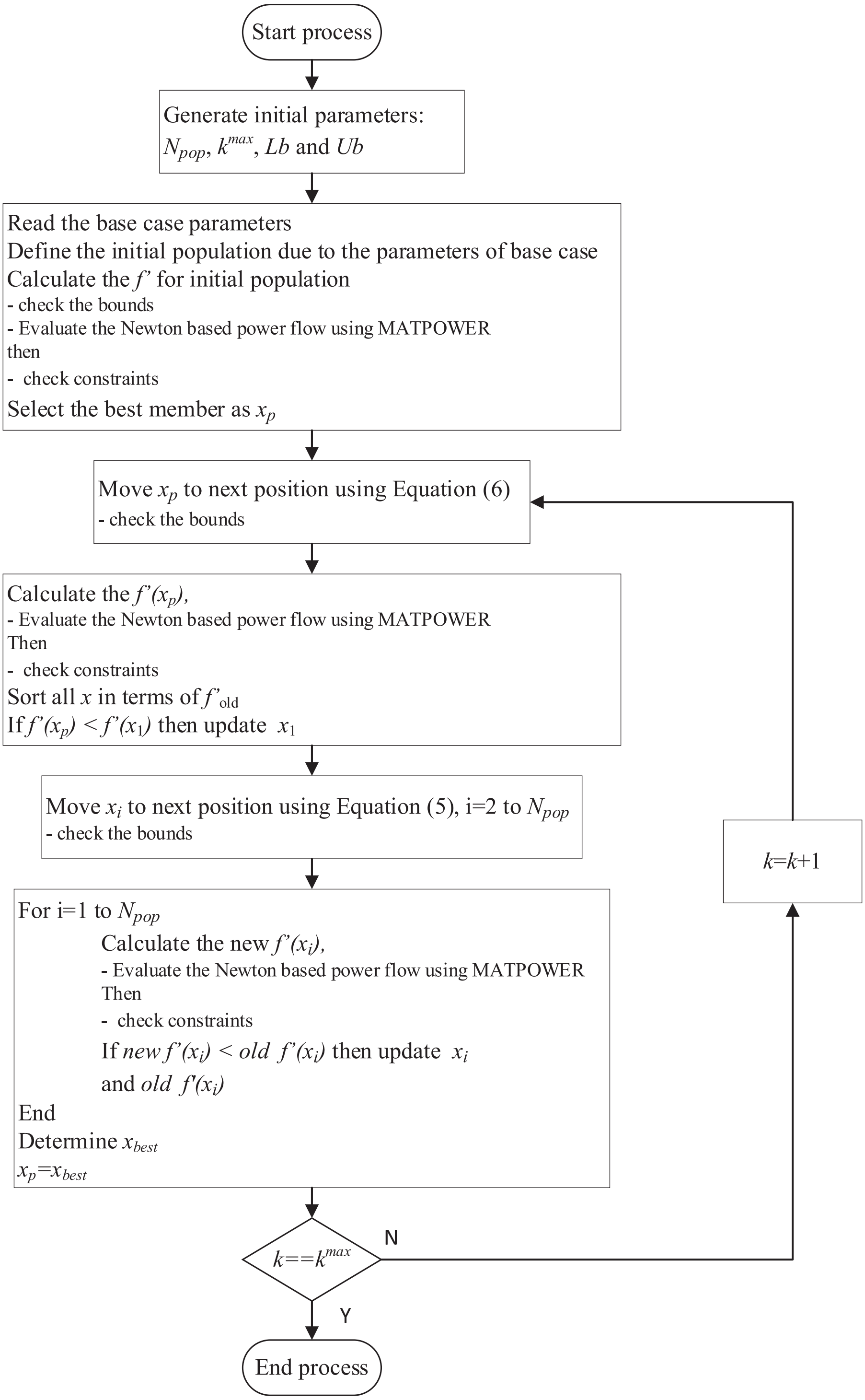

Figure 1. Flowchart of the modified pathfinder algorithm (mPFA).

(1) Read the data and then specify the parameter: iteration number, population size, dimension of search space.
(2) Initialize the lower and upper bounds. Then generate the first population using base case data. Check the bounds; if variables lower then lower bound, they are fixed to lower bound and if variables higher then upper bound, they are fixed to upper bound. Calculate the total power loss of initial members with assigning the position vectors to input data of MATPOWER and run the power flow in MATPOWER. MATOPWER is executed for each position vector, separately. Then read the dependent variables. If they violate their bounds, they are added to solution as penalty terms formulated in Equation (5).
(3) $k = 0$
(4) While stopping criteria $\{k < kmax\}$, let $k = k + 1$
(5) Update parameter of modified PFA and generate A and ε.
(6) Move the pathfinder and other members toward next position using Equation (7) and Equation (6), respectively. Check the lower and upper bounds. if variables lower then lower bound, they are fixed to lower bound and if variables higher then upper bound, they are fixed to upper bound.
(7) Calculate the total power loss of initial members with assigning the position vectors to input data of MATPOWER and run the power flow in MATPOWER. MATOPWER is executed for each position vector, separately. Then read the dependent variables. If they violate their bounds, they are added to solution as penalty terms formulated in Equation (5).
(8) Update the population utilizing "*if then*" rules as follows; if $f'\left(x_i^{k+1}\right) < f'(x_i^k)$, then $f'(x_i^k) = f'\left(x_i^{k+1}\right)$ and $x_i^k = x_i^{k+1}$, $i \in [2, N_{pop}]$. Select new pathfinder in last population.
(9) Finally update the pathfinder as follows; if $f'\left(x_p^{k+1}\right) < f'(x_p^k)$, then $f'(x_p^k) = f'\left(x_p^{k+1}\right)$ and $x_p^k = x_p^{k+1}$.
(10) Take the variable vector of pathfinder as output and display the results.

Figure 2. Implementation of the modified pathfinder algorithm (mPFA).

5. Results of simulations

The simulations were performed on two IEEE test systems: 57-bus and 118-bus systems. These test beds are large-scale test systems that are frequently discussed in the literature. In particular, the 118-bus test system comprises a large number of control variables and thus it is a challenging test bed for the ORPD problem. To verify and show the performance and ability of the mPFA, the results are compared with some well-known methods proposed in the literature. The main objective is to minimize the total power losses. Moreover, parametric analysis and statistical tests such as Kolmogorov–Smirnov, Friedman and Wilcoxon tests are performed on 57-bus system. All algorithms are coded and run in MATLAB® 2016 on a personal computer with Intel i5 central processing unit (CPU) and 4 GB RAM. The data in MATPOWER were used.

5.1. Parametric analysis

The mPFA is implemented on the IEEE 57-bus system in four cases. In first, second, third and fourth cases, the number of search agents is set as 10, 30, 60 and 100, respectively. The convergence curve is shown in Figure 3. When it is analysed for each case, it can be seen that the proposed algorithm generates new solutions over the course of iterations. The elapsed times for the first run are 34 s in the first case, 98 s in the second case, 173 s in the third case and 311 s in the fourth case. The best results are achieved when the number of search agents is selected as 60 and 100. However, the fourth case is time expensive, since there are too many function evaluations (equal to 30,000). On the other hand, the worst results are achieved when the number of search agents is set as 10. When the number of search agents is set as 60 and 30, power loss is obtained as 24.3692 and 24.4190 MW, respectively. However, it would be more compatible to select the number of search agents as 30.

The deterministic parameter model (Yapici and Cetinkaya 2019), which uses some rules to adjust the parameters over the course of the process, was used.

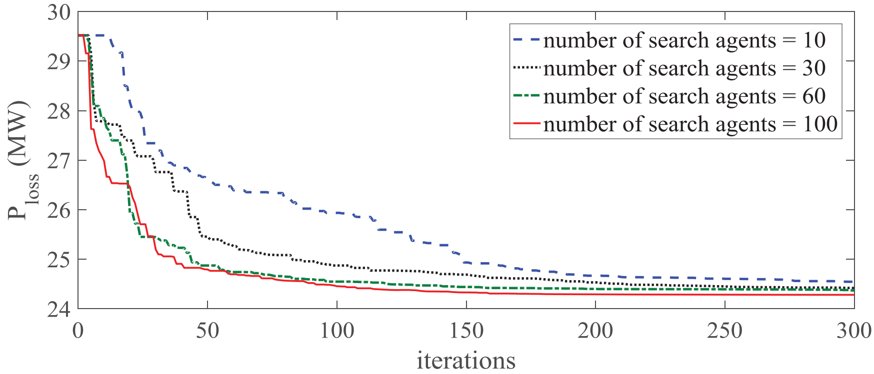

Figure 3. Convergence curves using different numbers of search agents.

5.2. Experimental analysis of the proposed modification

The PFA has two important adjustable parameters: A and ε. By adjusting the coefficients of these parameters, the search agents can move in the search space with certain steps. If the coefficients of A and $\varepsilon < 1$, then the search agents will move to the next position with small steps. If the coefficients of A and $\varepsilon > 1$, then the search agents will move towards the next position with large steps.

To produce new solutions, the position vectors in the search space need to be changed with small steps. Therefore, some modifications are required to guarantee the generation of new solutions. In this regard, some different modifications have been examined experimentally. Among the many possibilities, the four most effective modifications for comparison are covered in this subsection. These are as follows:

$$A = 0.1 \times A$$
$$A = 0.1 \times A, \varepsilon = 0.1 \times \varepsilon$$
$$A = 0.1 \times A, \varepsilon = 0.001 \times \varepsilon$$
$$A = 0.001 \times A, \varepsilon = 0.001 \times \varepsilon$$

All algorithms are tested on the 57-bus and 118-bus test systems. The proposed modification (mPFA) was compared with the original PFA and the modifications listed above. The results are given in Table 1. In addition, for 57-bus and 118-bus test systems, the convergence curves are shown in Figures 4 and 5. It can be observed that the original PFA was not very successful in producing new solutions. The proposed method achieved less power loss than the original PFA and other modifications. It was able to generate promising solutions over the course of the iterations. The mPFA obtained the best solutions as 22.6938 and 114.8970 MW for the 57-bus and 118-bus systems, respectively. For the 57-bus system, the mPFA improved the best optimal solution by up to 8.03% compared with the original PFA. For the 118-bus system, the mPFA improved the best optimal solution by up to 3.35% compared with the PFA. The CPU times given in Table 1 are almost equal. This means that the modifications do not change the general mechanism and complexity of the PFA.

5.3. Statistical tests

In this subsection, several statistical tests are performed to assess the compatibility of the results and rank the mPFA: the Kolmogorov–Smirnov test, Friedman test and Wilcoxon test. The mPFA is compared with several methods, listed below:

- GWO (Sulaiman *et al.* 2015)

Table 1. Performance testing of the modified pathfinder algorithm (mPFA) compared with the pathfinder algorithm (PFA) and other modifications.

Test system	Method	Minimum	Mean	Maximum	Std. dev.	CPU time (s)	Maximum no. of iterations	Population size	No. of function evaluations
IEEE 57-bus test system	Original PFA	24.6752	26.7835	27.8539	1.0033	14.895	300	30	9000
	First modification	24.2243	25.1276	26.7905	0.7133	14.927	300	30	9000
	Second modification	25.5512	26.2545	27.0094	0.3578	14.950	300	30	9000
	Third modification	25.8626	26.5386	27.0616	0.3077	14.903	300	30	9000
	Fourth modification	23.6653	24.2526	24.9721	0.3429	14.881	300	30	9000
	Proposed modification	22.6938	22.9187	23.2427	0.1124	14.895	300	30	9000
IEEE 118-bus test system	Original PFA	118.8741	124.1819	130.6500	3.4477	39.255	300	30	9000
	First modification	118.4335	122.6752	128.3496	2.2180	39.289	300	30	9000
	Second modification	118.7002	121.0619	123.9366	1.5193	39.403	300	30	9000
	Third modification	126.8512	128.2656	129.6920	0.6522	39.412	300	30	9000
	Fourth modification	118.8011	120.6754	122.1709	0.7930	39.300	300	30	9000
	Proposed modification	114.8970	116.1557	117.9215	0.6373	39.318	300	30	9000

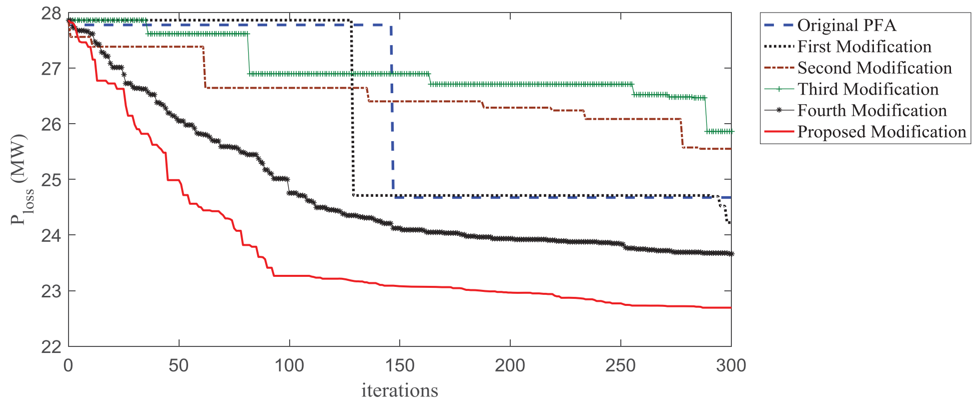

Figure 4. Convergence curves for the 57-bus system. PFA = pathfinder algorithm.

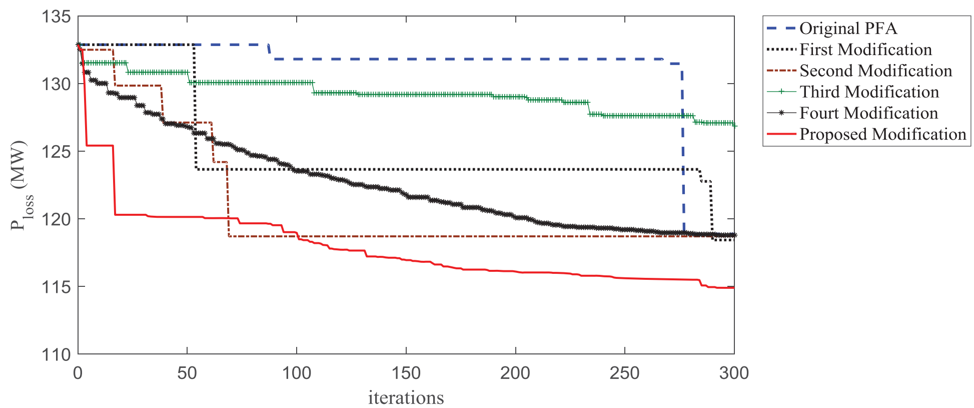

Figure 5. Convergence curves for the 118-bus system. PFA = pathfinder algorithm.

- MFO (Mei *et al.* 2017)
- teaching–learning-based artificial bee colony (TLABC) (Chen *et al.* 2018)
- symbiotic organisms search (SOS) (Cheng and Prayogo 2014)
- improved Jaya optimization algorithm (IJAYA) (Yu *et al.* 2017)
- e-JAYA (Barakat *et al.* 2019)
- ALO (Li *et al.* 2019)
- IALO (Li *et al.* 2019)
- PSO (Mahadevan and Kannan 2010)
- GA (Devaraj and Roselyn 2010).

For a reasonable comparison, the number of agents and number of maximum iterations are set as 30 and 300 for all methods, respectively. All methods are executed by running 30 free trials. (The parameters of the algorithms are listed in Table S1 in the Supplementary Material.) MFO, IJAYA and e-JAYA start with random populations owing to their structures. To examine the consistency of results obtained by the mPFA and to evaluate the distribution characteristics of these values, the Kolmogorov–Smirnov test is executed. According to this analysis, the e-JAYA method exhibited a normal distribution with an asymptotic-signum value of 0.000 $<$ 0.005. From the Kolmogorov–Smirnov test, the proposed method obtained an asymptotic-signum value of 0.935. PSO, GA, GWO, MFO, TLABC, SOS, IJAYA, ALO and IALO obtained asymptotic-signum values of 0.174, 0.071, 0.481, 0.249, 0.230, 0.414, 0.098, 0.958 and 0.976, respectively. Another test is the Friedman test, which was used to compare the proposed method with the other methods. This test is a multiple non-parametric analysis which emphasizes differences in statistical significance for at least one pair of competitor algorithms. The Friedman test had a chi-squared distribution with 270.690 degrees of freedom, with an asymptotic-signum value equal to 0.000. All methods were tested cumulatively. It can be observed that the available variables are meaningfully different from each other. As a result, it can be accepted that the mPFA achieves the best performance with a mean rank equal to 1.77. e-JAYA and IALO are ranked as second and third, with mean ranks equal to 1.93 and 3.97, respectively. PSO, GA, GWO, MFO, TLABC, SOS, IJAYA and ALO achieved mean rank values of 9.63, 12.83, 6.20, 9.10, 9.50, 7.40, 6.30, 4.07 and 3.97, respectively. The third test is the Wilcoxon test, which was carried out to compare the mPFA with other algorithms, separately. As the results of the comparison of mPFA and e-JAYA, the p values are obtained as $0.069 > 0.05$. This means that in terms of their performance, the two methods are statistically similar. In other words, although the mPFA shows a significant difference from other algorithms with a value of $0.000 < 0.005$, there is no significant difference from e-JAYA. However, the mPFA achieved a lower value of standard deviation than all other methods. This shows that it is more consistent than all the other methods.

5.4. Simulation for 57-bus system

The proposed method was compared to several existing methods to evaluate the performance on the 57-bus test system. The proposed algorithm uses the parameter settings introduced by Li *et al.* (2019). The performance of the mPFA was evaluated by adjusting the population size and number of iterations to 25 and 200, respectively. In the context of testing the mPFA, the numerical analysis was carried out by running 50 free executions.

For comparison, the results obtained by the proposed method, including performance indices such as minimum, mean and maximum power losses, standard deviation and CPU time, are given in Table 2. The results show that the mPFA outperformed many algorithms in solving the ORPD problem for the 57-bus test system. According to the results listed in Table 2, multi-objective GWO showed the best performance, with a result of 21.171 MW. Also, IALO outperformed the mPFA, with a result of 22.2539 MW. With respect to the results compared with other methods, it can be concluded that the mPFA has an efficient searching capability since it reduced the total power losses to 22.3450 MW. In the comparison of CPU time, the mPFA completed the first execution in 14.9553 s and ranked second. Consequently, it can be observed that the proposed method obtained competitive results in the analysis performed on the 57-bus test system. The mPFA improved the best solution more than DE, SOA, GA, PSO, GSA, ALC-PSO, MFO and ALO, with results of 10.79%, 7.91%, 12.85%, 10.73%, 4.75%, 4.47%, 4.30%, 7.87% and 2.37%, respectively. However, multi-objective GWO and IALO improved the best solution more than the mPFA, with results of 5.55% and 0.41%, respectively.

The optimal solutions of control variables obtained by the mPFA for the IEEE 57-bus system are tabulated in Table S2 in the Supplementary Material.

5.5. Simulation for 118-bus system

In this case study, the performance of the mPFA was tested by comparing the reduction of power loss with various existing methods. The proposed algorithm uses two different parameter settings, with limitations of the control variables explained below.

- **Case 1**: mPFA uses the parameters reported by Mei *et al.* (2017), which are lower bounds, upper bounds, number of search agents, maximum number of iterations and total number of function evaluations.

Table 2. Results for IEEE 57-bus system.

Method	Minimum P_{Loss} (MW)	Mean P_{Loss} (MW)	Maximum P_{Loss} (MW)	Std. dev.	CPU time (s)	Maximum no. of iterations	Population size	No. of function evaluations
DE (Varadarajan and Swarup 2008)	25.0475	25.1112	25.2016	0.049	35.654	200	30	6000
SOA (Dai *et al.*2009)	24.26548	24.27078	24.28046	0.0042	–	300	60	18,000
GA (Khazali and Kalantar 2011)	25.64	26.8378	27.7651	–	–	–	–	–
PSO (Khazali and Kalantar 2011)	25.03	26.4742	27.0576	–	–	–	–	–
GSA (Duman *et al.*2012)	23.46	–	–	–	321.4872	150	90	–
ALC-PSO (Singh, Mukherjee, and Ghoshal 2015)	23.39	24.41	–	–	300.78	500	60	30,000
MFO (Mei *et al.*2017)	24.25293	–	–	–	–	300	30	9000
Multi-objective GWO (Li *et al.*2019)	21.171	–	–	–	–	100	30	6000
ALO (Li *et al.*2019)	22.8884	23.5584	–	–	20.40	200	25	5000
IALO (Li *et al.*2019)	22.2539	23.54293	–	–	14.75	200	25	5000
mPFA	22.3450	22.9116	23.2192	0.1160	14.9553	200	25	5000

Note: DE = differential evolution; SOA = seeker optimization algorithm; GA = genetic algorithm; PSO = particle swarm optimization; GSA = gravitational search algorithm; ALC-PSO = particle swarm optimization with an ageing leader and challengers; MFO = moth–flame optimizer; GWO = grey wolf optimizer; ALO = ant lion optimizer; IALO = improved ant lion optimizer.

- **Case 2**: mPFA uses the parameters reported by Li *et al.* (2019), which are lower bounds, upper bounds, number of search agents, maximum number of iterations and total number of function evaluations.

The results are given in Table 3. In the first case, the results were obtained in 30 free executions, while the results in the second case were obtained in 50 free executions. For the first and second cases, the mPFA obtained the power loss as 117.0690 and 114.6092 MW, respectively. In addition, it completed the simulation in the CPU time of 420.852 s for case 1 and 39.308 s for case 2. MFO found a better optimal solution than mPFA in terms of considering the bounds for control variables and other parameters. In case 2, the mPFA obtained more effective and competitive results owing to the changing limits of control variables. However, ISSA and SDP found more effective and better results than the proposed method. On the other hand, the mPFA outperformed the other methods in the context of minimizing power losses.

CPU time is given as another performance criterion. SDP has a lower CPU time than all the other methods. On the other hand, the proposed method completed the optimization process in less time than many of the other algorithms. This proves that the proposed method is less complex.

Also, the mPFA improved the best solution more than DE, CLPSO, PSO, PSO-TVIW, PSO-TVAC, SPSO-TVAC, PSO-CF, PGS-PSO, SWT-PSO, PGSWT-PSO, PSO-IPGS, SOA, GWO, ALC-PSO, GSA, MFO, ALO, IALO, MSFS and case 1, with results of 10.68%, 12.49%, 13.17%, 1.96%, 7.82%, 1.37%, 0.90%, 1.71%, 7.68%, 4.03%, 0.39%, 0.30%, 5.01%, 5.69%, 10.29%, 1.56%, 1.93%, 0.16%, 0.01% and 2.10%, respectively. (These algorithm abbreviations are defined in the footnote to Table 3.) However, SDP and ISSA improved the best optimal solution more than mPFA, with results of 1.27% and 0.07%, respectively.

The results of control variables obtained by the mPFA for the two different cases are given in Table S3 in the Supplementary Material.

6. Discussion

From the results of the case studies, an advantage of the mPFA is that the successful transition between the exploration and exploitation phases enabled new solutions to be produced over the course of iterations, thus obtaining better results than many methods in solving the ORPD problem. In contrast, a disadvantage of the mPFA is that it produces solutions with minor changes towards the end of process, and thus, it cannot produce more influential solutions. The number of iterations becomes an important problem owing to the convergence of A and ε towards 0, and even finding promising solutions becomes difficult. Therefore, it is important to keep the number of iterations at a certain value. However, this may be time consuming. Accordingly, the number of iterations should be chosen in a small range. While this reduces the processing time, A and ε cannot be updated sufficiently. This may cause a deficiency in the iterative process. After many executions in all case studies, it was observed that the irregular movements of followers cause them to move away from promising solutions. But thanks to the leader individual using a completely different mathematical model, it was possible to obtain new solutions at the end of the process.

As a result, it was proved that the mPFA is a robust optimization algorithm, which has a superior ability to solve the ORPD problem while achieving effective and competitive results. Moreover, all control variables are within their limits.

7. Conclusion

In this article, a modified version of the pathfinder algorithm (mPFA) was implemented to solve the ORPD problem. The numerical analyses were performed on IEEE 57-bus and 118-bus test systems to minimize power losses. The mPFA achieved relatively competitive results in comparison with other well-known approaches. From the results obtained, the mPFA proved to be more effective and robust

Table 3. Results for IEEE 118-bus system.

Method	Minimum P_{Loss} (MW)	Mean P_{Loss} (MW)	Maximum P_{Loss} (MW)	Std. dev.	CPU time (s)	Maximum no. of iterations	Population size	No. of function evaluations
DE (Varadarajan and Swarup 2008)	128.318	129.0817	129.5790	0.345	42.1556	–	30	–
CLPSO (Mahadevan and Kannan 2010)	130.96	–	132.74	–	–	200	120	–
PSO (Mahadevan and Kannan 2010)	131.99	–	134.5	–	–	200	120	–
PSO-TVIW (Polprasert, Ongsakul, and Dieu 2016)	116.8979	118.2344	126.6222	1.6009	109.645	200	40	8000
PSO-TVAC (Polprasert, Ongsakul, and Dieu 2016)	124.3335	129.7494	134.1254	2.156	96.32	200	40	8000
SPSO-TVAC (Polprasert, Ongsakul, and Dieu 2016)	116.2026	117.3553	118.139	0.4696	96.45	200	40	8000
PSO-CF (Polprasert, Ongsakul, and Dieu 2016)	115.6469	116.9863	119.8378	0.8655	95.86	200	40	8000
PGS-PSO (Polprasert, Ongsakul, and Dieu 2016)	116.6075	119.3968	127.0772	2.107	96.11	200	40	8000
SWT-PSO (Polprasert, Ongsakul, and Dieu 2016)	124.1476	129.371	141.6147	3.309	91.58	200	40	8000
PGSWT-PSO (Polprasert, Ongsakul, and Dieu 2016)	119.427	122.781	125.762	1.2455	95.17	200	40	8000
PSO-IPGS (Polprasert, Ongsakul, and Dieu 2016)	115.06	116.462	118.35	0.528	91.07	200	40	8000
SOA (Dai *et al.*2009)	114.95013	116.34725	115.67443	0.0035908	–	300	60	18,000
GWO (Sulaiman *et al.*2015)	120.65	–	–	–	–	150	40	6000
ALC-PSO (Singh, Mukherjee, and Ghoshal 2015)	121.53	–	132.99	–	34.84	–	60	500
GSA (Duman *et al.*2012)	127.76	–	–	–	39.95	150	90	–
MFO (Mei *et al.*2017)	116.4254	–	–	–	–	1000	30	30,000
ISSA (Nguyen and Vo 2019)	114.5297	115.651	121.1127	1.4889	41.6	150	40	7190
ALO (Li *et al.*2019)	116.86	119.712	–	–	50.71	250	30	7500
IALO (Li *et al.*2019)	114.795	117.299	–	–	39.59	250	30	7500
MSFS (Nguyen *et al.*2019)	114.6251	115.4278	116.6677	0.4678	63.7	200	15	9000
SDP (Davoodi *et al.*2019)	113.17	–	–	–	6.07	–	–	–
$\mathrm{mPFA}_{\mathrm{Case1}}$	117.0690	117.3823	118.0053	0.2969	420.852	1000	30	30,000
$\mathrm{mPFA}_{\mathrm{Case2}}$	114.6092	116.1976	116.9854	0.6910	39.308	250	30	7500

Note: DE = differential evolution; CLPSO = comprehensive learning particle swarm optimization; PSO = particle swarm optimization; PSO-TVIW = particle swarm optimization with time-varying inertia weight; PSO-TVAC = particle swarm optimization with time-varying acceleration coefficients; SPSO-TVAC = self-organizing hierarchical particle swarm optimization with time-varying acceleration coefficients; PSO-CF = particle swarm optimization with constriction factor; PGS-PSO = pseudo-gradient search particle swarm optimization; SWT-PSO = particle swarm optimization with stochastic weight trade-off; PGSWT-PSO = pseudo-gradient particle swarm optimization with stochastic weight trade-off; PSO-IPGS = improved pseudo-gradient search particle swarm optimization; SOA = seeker optimization algorithm; GWO = grey wolf optimizer; ALC-PSO = particle swarm optimization with an ageing leader and challengers; GSA = gravitational search algorithm; MFO = moth–flame optimizer; ISSA = improved social spider algorithm; ALO = ant lion optimizer; IALO = improved ant lion optimizer; MSFS = modified stochastic search algorithm; SDP = semi-definite programming.

than the original PFA. In addition, for the 57-bus system, the mPFA improved the objective corresponding to the reduction of power losses by 8.03% more than the original PFA. Also, for the 118-bus system, the improvement of the mPFA over the original version was about 3.35%. According to the results of comparisons with existing methods, the mPFA proved its capability and superiority since it outperformed many methods by obtaining optimal solutions with better quality. For the 57-bus system and 118-bus system, the mPFA achieved the improvement level from 2.37% to 12.85% and from 0.01% to 13.17%, respectively, in minimizing power losses. In addition, the proposed method reached the end of the process in less time than many of the other algorithms. As a result, it can be seen that the proposed method obtained the control variables within acceptable values by effectively addressing the equality and inequality constraints.

The statistical tests also showed that the results obtained by the mPFA are consistent. According to the Kolmogorov test, to emphasize the consistency, the mPFA ranked as second, with a result equal to 0.935. This showed that the PFA was more consistent than many other methods. The mPFA obtained better performance than all other methods, with a mean rank equal to 1.77 in the Friedman test. Finally, according to the Wilcoxon test, although the mPFA was statistically similar to e-JAYA, it differed significantly from the other methods, with a *p*-value of 0.000. Furthermore, the mPFA obtained the best standard deviation. This means that it is more consistent than the other methods.

The mPFA could be used for other power system problems such as economic power dispatch, system reconfiguration and optimal design problems. To improve the performance of the proposed method, other approaches could be used for hybridization using their superior features. In future work, the mPFA will be used to find optimal weights of artificial neural networks in order to control a battery management system for standalone photovoltaic units.

Disclosure statement

No potential conflict of interest was reported by the author.

Funding

This research is not supported by any Foundation or Corporation.

ORCID

Hamza Yapici http://orcid.org/0000-0003-0687-2953

References

Abdel-Fatah, S., M. Ebeed, and S. Kamel. 2019. "Optimal Reactive Power Dispatch Using Modified Sine Cosine Algorithm." In *Proceedings of 2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, 510–514. Piscataway, NJ: IEEE..

Awad, N. H., M. Z. Ali, R. Mallipeddi, and P. N. Suganthan. 2019. "An Efficient Differential Evolution Algorithm for Stochastic OPF Based Active–Reactive Power Dispatch Problem Considering Renewable Generators." *Applied Soft Computing* 76: 445–458. doi:10.1016/j.asoc.2018.12.025.

Ayan, K., and U. Kılıc. 2012. "Artificial Bee Colony Algorithm Solution for Optimal Reactive Power Flow." *Applied Soft Computing* 12 (5): 1477–1482. doi:10.1016/j.asoc.2012.01.006.

Barakat, A. F., R. A. El-Sehiemy, M. Elsaid, and E. Osman. 2018. "Solving Reactive Power Dispatch Problem by Using JAYA Optimization Algorithm." *International Journal of Engineering Research in Africa* 36: 12–24. doi:10.4028/www.scientific.net/JERA.36.12.

Barakat, A. F., R. A. El-Sehiemy, M. I. Elsayd, and E. Osman. 2019. "An Enhanced Jaya Optimization Algorithm (EJOA) for Solving Multi-objective ORPD Problem." In *Proceedings of 2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, 479–484. Piscataway, NJ: IEEE.

Bingane, C., M. F. Anjos, and S. Le Digabel. 2019. "Tight-and-Cheap Conic Relaxation for the Optimal Reactive Power Dispatch Problem." *IEEE Transactions on Power Systems* 34 (6): 4684–4693. doi:10.1109/TPWRS.2019.2912889.

Biswas, P. P., P. N. Suganthan, R. Mallipeddi, and G. A. Amaratunga. 2019. "Optimal Reactive Power Dispatch with Uncertainties in Load Demand and Renewable Energy Sources Adopting Scenario-Based Approach." *Applied Soft Computing* 75: 616–632. doi:10.1016/j.asoc.2018.11.042.

Chen, X., B. Xu, C. Mei, Y. Ding, and K. Li. 2018. "Teaching–Learning-Based Artificial Bee Colony for Solar Photovoltaic Parameter Estimation." *Applied Energy* 212: 1578–1588. doi:10.1016/j.apenergy.2017.12.115.

Cheng, M. Y., and D. Prayogo. 2014. "Symbiotic Organisms Search: A New Metaheuristic Optimization Algorithm." *Computers and Structures* 139: 98–112. doi:10.1016/j.compstruc.2014.03.007.

Dai, C., W. Chen, Y. Zhu, and X. Zhang. 2009. "Seeker Optimization Algorithm for Optimal Reactive Power Dispatch." *IEEE Transactions on Power Systems* 24 (3): 1218–1231. doi:10.1109/TPWRS.2009.2021226.

Das, T., R. Roy, K. K. Mandal, S. Mondal, S. Mondal, P. Hait, and M. K. Das. 2020. "Optimal Reactive Power Dispatch Incorporating Solar Power Using Jaya Algorithm." In *Computational Advancement in Communication Circuits and Systems*, edited by K. Maharatna, M. Kanjilal, S. Konar, S. Nandi, and K. Das, 37–48. Singapore: Springer.

Davoodi, E., E. Babaei, B. Mohammadi-Ivatloo, and M. Rasouli. 2019. "A Novel Fast Semidefinite Programming-Based Approach for Optimal Reactive Power Dispatch." *IEEE Transactions on Industrial Informatics* 16 (1): 288–298.

Devaraj, D., and J. P. Roselyn. 2010. "Genetic Algorithm Based Reactive Power Dispatch for Voltage Stability Improvement." *International Journal of Electrical Power and Energy Systems* 32 (10): 1151–1156. doi:10.1016/j.ijepes.2010.06.014.

Duman, S., U. Güvenc, Y. Sönmez, and N. Yörükeren. 2012. "Optimal Power Flow Using Gravitational Search Algorithm." *Energy Conversion and Management* 59: 86–95. doi:10.1016/j.enconman.2012.02.024.

Durairaj, S., D. Devaraj, and P. S. Kannan. 2006. "Genetic Algorithm Applications to Optimal Reactive Power Dispatch with Voltage Stability Enhancement." *Journal-Institution of Engineers India Part El Electrical Engineering Division* 87: 42–47.

Ghasemi, M., M. M. Ghanbarian, S. Ghavidel, S. Rahmani, and E. M. Moghaddam. 2014. "Modified Teaching Learning Algorithm and Double Differential Evolution Algorithm for Optimal Reactive Power Dispatch Problem: A Comparative Study." *Information Sciences* 278: 231–249. doi:10.1016/j.ins.2014.03.050.

Heidari, A. A., R. A. Abbaspour, and A. R. Jordehi. 2017. "Gaussian Bare-Bones Water Cycle Algorithm for Optimal Reactive Power Dispatch in Electrical Power Systems." *Applied Soft Computing* 57: 657–671. doi:10.1016/j.asoc.2017.04.048.

Khazali, A. H., and M. Kalantar. 2011. "Optimal Reactive Power Dispatch Based on Harmony Search Algorithm." *International Journal of Electrical Power and Energy Systems* 33 (3): 684–692. doi:10.1016/j.ijepes.2010.11.018.

Le Dinh, L., D. Vo Ngoc, and P. Vasant. 2013. "Artificial Bee Colony Algorithm for Solving Optimal Power Flow Problem." *The Scientific World Journal* 2013 (5): 159040. doi:10.1155/2013/159040.

Li, Z., Y. Cao, L. V. Dai, X. Yang, and T. T. Nguyen. 2019. "Finding Solutions for Optimal Reactive Power Dispatch Problem by a Novel Improved Antlion Optimization Algorithm." *Energies* 12 (15): 2968. doi:10.3390/en12152968.

Mahadevan, K., and P. S. Kannan. 2010. "Comprehensive Learning Particle Swarm Optimization for Reactive Power Dispatch." *Applied Soft Computing* 10 (2): 641–652. doi:10.1016/j.asoc.2009.08.038.

Medani, K. B., S. Sayah, and A. Bekrar. 2018. "Whale Optimization Algorithm Based Optimal Reactive Power Dispatch: A Case Study of the Algerian Power System." *Electric Power Systems Research* 163: 696–705. doi:10.1016/j.epsr.2017.09.001.

Mei, R. N. S., M. H. Sulaiman, Z. Mustaffa, and H. Daniyal. 2017. "Optimal Reactive Power Dispatch Solution by Loss Minimization Using Moth-Flame Optimization Technique." *Applied Soft Computing* 59: 210–222. doi:10.1016/j.asoc.2017.05.057.

Mouassa, S., T. Bouktir, and A. Salhi. 2017. "Ant Lion Optimizer for Solving Optimal Reactive Power Dispatch Problem in Power Systems." *Engineering Science and Technology* 20: 885–895. doi:10.1016/j.jestch.2017.03.006.

Mukherjee, A., and V. Mukherjee. 2016. "Chaotic Krill Herd Algorithm for Optimal Reactive Power Dispatch Considering FACTS Devices." *Applied Soft Computing* 44: 163–190. doi:10.1016/j.asoc.2016.03.008.

Nguyen, T. T., T. T. Nguyen, L. T. Duong, and V. A. Truong. 2020. "An Effective Method to Solve the Problem of Electric Distribution Network Reconfiguration Considering Distributed Generations for Energy Loss Reduction." *Neural Computing and Applications*. doi:10.1007/s00521-020-05092-2.

Nguyen, T. T., and D. N. Vo. 2019. "Improved Social Spider Optimization Algorithm for Optimal Reactive Power Dispatch Problem with Different Objectives." *Neural Computing and Applications* 31: 1–32. doi:10.1007/s00521-019-04073-4.

Nguyen, T. T., D. N. Vo, H. Van Tran, and L. Van Dai. 2019. "Optimal Dispatch of Reactive Power Using Modified Stochastic Fractal Search Algorithm." *Complexity* 2019: Article ID 4670820. doi:10.1155/2019/4670820.

Pesaran, M. H., and A. Timmermann. 1992. "A Simple Nonparametric Test of Predictive Performance." *Journal of Business and Economic Statistics* 10 (4): 461–465. doi:10.1080/07350015.1992.10509922.

Polprasert, J., W. Ongsakul, and V. N. Dieu. 2016. "Optimal Reactive Power Dispatch Using Improved Pseudo-Gradient Search Particle Swarm Optimization." *Electric Power Components and Systems* 44: 518–532. doi:10.1080/15325008.2015.1112449.

Priyadarshani, S., K. R. Subhashini, and J. K. Satapathy. 2020. "Pathfinder Algorithm Optimized Fractional Order Tilt-Integral-Derivative (FOTID) Controller for Automatic Generation Control of Multi-source Power System." *Microsystem Technologies*. doi:10.1007/s00542-020-04897-4.

Qi, X., Z. Yuan, and Y. Song. 2020. "A Hybrid Pathfinder Optimizer for Unconstrained and Constrained Optimization Problems." *Computational Intelligence and Neuroscience* 2020 (7): 5787642. doi:10.1155/2020/5787642.

Saddique, M. S., A. R. Bhatti, S. S. Haroon, M. K. Sattar, S. Amin, I. A. Sajjid, S. S. ul-Haq, A. B. Awad, and N. Rasheed. 2020. “Solution to Optimal Reactive Power Dispatch in Transmission System Using Meta-heuristic Techniques—Status and Technological Review.” *Electric Power Systems Research* 178: 106031.

Shaheen, A. M., R. A. El-Sehiemy, and S. M. Farrag. 2018. “Integrated Strategies of Backtracking Search Optimizer for Solving Reactive Power Dispatch Problem.” *IEEE Systems Journal* 12: 424–433. doi:10.1109/JSYST.2016.2573799.

Simard, R., and P. L’Ecuyer. 2011. “Computing the Two-Sided Kolmogorov-Smirnov Distribution.” *Journal of Statistical Software* 39 (11): 1–18. doi:10.18637/jss.v039.i11.

Singh, R. P., V. Mukherjee, and S. P. Ghoshal. 2015. “Optimal Reactive Power Dispatch by Particle Swarm Optimization with an Aging Leader and Challengers.” *Applied Soft Computing* 29: 298–309. doi:10.1016/j.asoc.2015.01.006.

Sulaiman, M. H., Z. Mustaffa, M. R. Mohamed, and O. Aliman. 2015. “Using the Gray Wolf Optimizer for Solving Optimal Reactive Power Dispatch Problem.” *Applied Soft Computing* 32: 286–292. doi:10.1016/j.asoc.2015.03.041.

Üney, M. Ş., and N. Cetinkaya. 2019. “New Metaheuristic Algorithms for Reactive Power Optimization.” *Tehnički Vjesnik* 26 (5): 1427–1433. doi:10.17559/TV-20181205153116.

Varadarajan, M., and K. S. Swarup. 2008. “Differential Evolutionary Algorithm for Optimal Reactive Power Dispatch.” *International Journal of Electrical Power and Energy Systems* 30 (8): 435–444. doi:10.1016/j.ijepes.2008.03.003.

Wang, H., C. E. Murillo-Sanchez, R. D. Zimmerman, and R. J. Thomas. 2007. “On Computational Issues of Market-Based Optimal Power Flow.” *IEEE Transactions on Power Systems* 22 (3): 1185–1193. doi:10.1109/TPWRS.2007.901301.

Wu, Q. H., and J. T. Ma. 1995. “Power System Optimal Reactive Power Dispatch Using Evolutionary Programming.” *IEEE Transactions on Power Systems* 10 (3): 1243–1249. doi:10.1109/59.466531.

Yapici, H., and N. Cetinkaya. 2019. “A New Meta-heuristic Optimizer: Pathfinder Algorithm.” *Applied Soft Computing* 78: 545–568. doi:10.1016/j.asoc.2019.03.012.

Yapıcı, H., and N. Çetinkaya. 2017. “An Improved Particle Swarm Optimization Algorithm Using Eagle Strategy for Power Loss Minimization.” *Mathematical Problems in Engineering* 2017: 1. doi:10.1155/2017/1063045.

Yu, K., J. J. Liang, B. Y. Qu, X. Chen, and H. Wang. 2017. “Parameters Identification of Photovoltaic Models Using an Improved JAYA Optimization Algorithm.” *Energy Conversion and Management* 150: 742–753. doi:10.1016/j.enconman.2017.08.063.

Yuan, Z., H. Li, and N. Yousefi. 2021. “Optimal Hydrogen Consumption of Fuel Cell-Based Locomotive Using Speed Trajectory Optimization by Improved Pathfinder Algorithm.” *Journal of Cleaner Production* 278: 123430. doi:10.1016/j.jclepro.2020.123430.

Zandi, Z., E. Afjei, and M. Sedighizadeh. 2012. “Reactive Power Dispatch Using Big Bang-Big Crunch Optimization Algorithm for Voltage Stability Enhancement.” In *2012 IEEE International Conference on Power and Energy*, 239–244. Piscataway, NJ: IEEE.

Zhao, B., C. X. Guo, and Y. J. Cao. 2005. “A Multiagent-Based Particle Swarm Optimization Approach for Optimal Reactive Power Dispatch.” *IEEE Transactions on Power Systems* 20 (2): 1070–1078. doi:10.1109/TPWRS.2005.846064.

Zimmerman, R. D., C. E. Murillo-Sánchez, and R. J. Thomas. 2010. “MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education.” *IEEE Transactions on Power Systems* 26 (1): 12–19. doi:10.1109/TPWRS.2010.2051168.

Zimmerman, D. W., and B. D. Zumbo. 1993. “Relative Power of the Wilcoxon Test, the Friedman Test, and Repeated-Measures ANOVA on Ranks.” *The Journal of Experimental Education* 62 (1): 75–86. doi:10.1080/00220973.1993.9943832.