A Phase-Based Adaptive Differential Evolution Algorithm for the Economic Load Dispatch Considering Valve-Point Effects and Transmission Losses

Xin Shen, Dexuan Zou, Xin Zhang, Qiang Zhang, and Peng Xiao

School of Electrical Engineering and Automation, Jiangsu Normal University, Xuzhou 221116, China

Correspondence should be addressed to Dexuan Zou; zoudexuan@163.com

Received 10 August 2018; Revised 22 October 2018; Accepted 29 October 2018; Published 15 November 2018

Copyright © 2018 Xin Shen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A phase-based adaptive differential evolution (PADE) algorithm is proposed to solve the economic load dispatch (ELD) considering valve-point effects (VPE) and transmission losses. To a great extent, PADE makes up for the drawbacks of the traditional differential evolution (DE) through three improvements. First, we establish an archive of storing successful individuals to improve the quality of offspring. Second, to balance the exploring and exploiting ability of the algorithm, a phase-based mutation operation is carried out. Third, two control parameters are adaptively adjusted, which is helpful for enhancing the robustness of the algorithm. In addition, two types of repair methods of constraint handling are employed for the ELD without or with transmission losses to help PADE find feasible solutions more efficiently. A performance comparison between PADE and other DE approaches from the literature was carried out on six ELD test cases which consider a set of operating constraints including the VPE and transmission losses. Results show a competitive PADE performance in all test cases regarding the compared DE approaches. Compared to methods from the literature, the costs obtained by PADE are lower in most cases while the corresponding constraint violations reach a lower level.

1. Introduction

In power systems, economic load dispatch (ELD) [1] is an important, complex optimization problem that allocates the required generation among the available generators. The main purpose of ELD is to find the optimal distribution of the generating units to make the fuel cost minimize while meeting all inequality and equality constraints in practice. Therefore, an accurate mathematical model is needed for adapting the ELD problems with different characteristics. So far, the quadratic function has been widely used as the classical mathematical model for solving the cost of the ELD problems. Moreover, addition conditions are often considered in the actual system, which constitute various versions for the ELD problems such as the valve-point effects (VPE) [2], transmission losses [3], and prohibited operating zones (POZs) [4].

Many classical methods, such as linear programming [5], quadratic programming [6], lambda iteration method [7], and gradient method [1], have been applied to solve the conventional ELD problems with the quadratic function. The above classical methods require that the generators exhibit convex and smooth characteristics. However, the input-output characteristics of generators are generally nonconvex and non-smooth in practical thermal generation plants. The reasons are caused by the VPE, POZs and so on, which may make the cost function produce a number of local optima. Therefore, these classical methods can hardly find the global optima for such types of problems. Dynamic programming [8] is different from the above classical methods. It can handle nonconvex and nonsmooth ELD problems, but it is restricted by the curse of dimensionality that inevitably leads to extra computational cost. Thus it is not also a suitable one for practical ELD problems.

In order to seek the optimal solution of ELD, more and more scholars tend to use modern intelligent algorithms. These modern intelligent algorithms have huge advantage over classical methods, especially for the ELD problems with various operation constraints. They are essentially approximation methods, which do not need the derivative
information of the ELD problems. The practical ELD problems are quite complex, so most of scholars are committed to continuously improving or hybrid modern intelligent algorithms and developing new constraint handling mechanisms at present. Qin and Cheng et al. [9] adopted an orthogonal designed method and proposed auxiliary vector generation based on multiple strategies to enhance the effectiveness of orthogonal designed operation. Based on the adaptive adjustment of acceleration coefficients, the algorithm's robustness and global search capability can be improved by employing a tent chaotic map. Furthermore, a repair method is designed to handle the practical constraints. Coelho and Mariani [10] put forward an improved harmony search (IHS) algorithm based on exponential distribution for the ELD problems. In IHS, a repair process is adopted to find feasible solutions. Al-Betar and Awadallah et al. [11] developed a modified version of harmony search algorithm that is called tournament-based harmony search (THS) algorithm, where the tournament selection process replaces the random selection process in the memory consideration operator. The tournament selection process is beneficial to improve the convergence performance of the algorithm. The introduced repair strategy can completely guarantee the feasibility of each generated solution before calculating the fuel cost. Therefore, to some extent, the THS algorithm can achieve better solutions compared to many state-of-the-art optimization approaches. Mandal and Roy et al. [12] proposed a new krill herd algorithm (KHA) to solve the ELD problems with various constraints. The crossover and mutation operations of DE are embedded into the KHA algorithm. The four versions of KHA are presented to solve different scale ELD problems. Zou and Li et al. [13] proposed a new global particle swarm optimization (NGPSO) algorithm for the economic emission dispatch with or without transmission losses. First, NGPSO devises a new position updating equation that utilizes the global best particle to guide the convergence of all particles. Second, the randomization based on the uniform distribution is used to avoid the premature convergence in the late optimization process. Simultaneously, different treatment methods are adopted for the equality and inequality constraints, respectively. For the inequality constraints, a common and simple penalty function is used. More importantly, the equality constraints are handled by three important steps, where the equality constraints are completely desirable. Different from the reported variable selection [14, 15], a variable is randomly selected to treat all variables equally in the repair process. As a result, a large number of high-quality feasible solutions can be obtained. Kumar and Chaturvedi [16] solved the ELD problems using a fuzzy particle swarm optimization (FPSO). The quadratic cost function with valve point loading and prohibited operating zone is studied in the models. The suitable value of inertia weight is yielded by a fuzzy logic controller (FLC). The FLC is composed of a knowledge base, a fuzzification interface, an inference system, and the defuzzification interface. As a result, FPSO efficiently avoids premature convergence. Mahadad and Srairi [17] adopted a hybrid method based on genetic algorithm (GA), differential evolution (DE), and pattern search (PS) [18], namely, GA-DE-PS. First, GA and DE are jointly used to find the optimal solutions. Second, the optimal solutions achieved by GA and DE are adjusted by PS. Third, the new solutions obtained by PS are employed to GA and DE for the new initial solutions of GA and DE. The interactive mechanism allows individuals to communicate with each other, which is very helpful to balance the exploiting and exploring capability of GA-DE-PS. Jedd and Vahidinasab [19] proposed a modified harmony search algorithm (MHSA). In MHSA, a new improving method based on wavelet mutation combined with a new memory consideration scheme based on the roulette wheel mechanism is adopted, which improves the accuracy, accelerates convergence speed, and enhances robustness of the classical HSA. For the constraint handling strategy, a common penalty function method is used to find feasible solutions. Singh and Dhillon [20] proposed an opposition-based greedy heuristic search (OGHS) to solve practical optimization problems. The initial population is composed of several randomly generated candidate individuals based on the uniform distribution, and its opposite population is formed by the opposite-based learning strategy [21]. Then the better half of the two populations is adopted to carry out subsequent optimization process, which improves the convergence speed of the algorithm. The opposition-based learning is also applied to its migration to maintain the diversity. The mutation strategy applied is based on the greediness and randomness to find the global best solution.

According to the review above, most works are focused on enhancing or hybridizing intelligent algorithms, and some of them studied the constraint handling improvement for practical ELD problems. The inequality constraints of ELD problems are relatively easy to meet. In contrast, the equality constraints of the ELD problems are hard to handle. Thus they mainly study the treatment method of the equality constraints for the ELD problems. However, their methods of treating the equality constraint with transmission losses are usually the same as those without transmission losses. In fact, the complexity of both equality constraints without transmission losses and equality constraints with transmission losses has a great difference. Therefore, the method of handling the equality constraint with transmission losses should be different from that of handling the equality constraint without transmission losses, which can differently repair the infeasible solutions. For our studies, the equality constraint refers to the power balance constraint. Meanwhile, in order to further improve the performance of the original DE algorithm and repair the infeasible solutions, the contributions of this paper contain two aspects. First, a phase-based adaptive differential evolution algorithm (PADE) is proposed to improve the performance of the original DE algorithm. Second, two repair methods are used to properly handle the equality constraints of two types of ELD problems, respectively.

The remainder of the paper is organized as follows: Section 2 describes the mathematical model of the ELD problems; Section 3 introduces the traditional differential evolution algorithm and the proposed DE variant; simulation results and the corresponding comprehensive analysis are presented in Section 4; Section 5 concludes the paper.
2. Economic Load Dispatch (ELD)

2.1. Objective Function. The objective of ELD is to minimize the total fuel cost of the entire power system under the condition of meeting various constraints. The cost of ELD can be described as follows:

\[ \min f_c = \sum_{j=1}^{N_G} F_j \left( P_j \right) \]  

(1)

where \( f_c \) represents the total fuel cost of the power generating (in $/h). $/h represents the total fuel cost of the power generating per hour. \( F_j \) represents the cost function of the generating unit \( j \) (in MW). MW means megawatt. \( N_G \) is the number of units in the power system.

In general, the traditional cost function of ELD is represented as the quadratic function, which can be given by Coelho and Mariani [10]:

\[ F_j \left( P_j \right) = a_j P_j^2 + b_j P_j + c_j \]

(2)

where \( a_j, b_j, \) and \( c_j \) denote the cost coefficients of unit \( j \). Under most circumstances, the valve-point effects (VPE) are contained in the practical power system. In view of this condition, the sinusoidal component is combined with the quadratic function, and the practical cost function can be amended as follows [10]:

\[ F_j \left( P_j \right) = a_j P_j^2 + b_j P_j + c_j + \left| e_j \sin \left( f_j \left( P_j^{\text{min}} - P_j \right) \right) \right| \]

(3)

where \( e_j \) and \( f_j \) denote the cost coefficients of unit \( j \) considering VPE.

2.2. Constraints. The ELD problems are solved subject to some inequality and equality constraints, including power output limits, power balance, and prohibited operating zones (POZs).

2.2.1. Power Output Limits. The power output of each unit should be within its upper and lower bounds [10] in the power system.

\[ P_j^{\text{min}} \leq P_j \leq P_j^{\text{max}} \]

(4)

where \( P_j^{\text{min}} \) (in MW) and \( P_j^{\text{max}} \) (in MW) stand for the minimum and maximum power output of unit \( j \), respectively.

2.2.2. Power Balance Constraints. The total generated power should be equal to the sum of the total load demand and transmission losses [10].

\[ \sum_{j=1}^{N_G} P_j = P_D + P_L \]

(5)

where \( P_D \) denotes the total load demand of power system (in MW); \( P_L \) denotes the total transmission losses (in MW), and it can be commonly obtained by Kron’s loss formula as follows [13]:

\[ P_L = \sum_{j=1}^{N_G} \sum_{h=1}^{N_G} P_j B_{jh} P_h + \sum_{j=1}^{N_G} B_{0j} P_j + B_{00} \]

(6)

where \( B_{jh} \) denotes the transmission loss coefficient. Additionally, the simplified version of power balance constraint has been commonly studied by scholars. This version does not include the transmission loss \( P_L \), namely, \( P_L = 0 \). Furthermore, it can be expressed by

\[ \sum_{j=1}^{N_G} P_j = P_D \]

(7)

2.2.3. Prohibited Operating Zones (POZs). Under certain circumstances, a generating unit may not operate within certain ranges known as prohibited operating zones due to physical operation limitations. As a result, the feasible operation zones of the generating unit may be discontinuous. At this time, the feasible power output ranges of the generating unit can be properly described as follows [4]:

\[
\begin{align*}
\frac{P_{j,1}}{P_{j,k-1}} & \leq P_j \leq \frac{P_{j,k}}{P_{j,1}} \\
\frac{P_{j,1}}{P_{j,\text{NPZ},j}} & \leq P_j \leq \frac{P_{j,k}}{P_{j,\text{NPZ},j}} \\
\end{align*}
\]

(8)

where \( P_{j,k} \) (in MW) and \( P_{j,1} \) (in MW) denote the lower and upper bounds of prohibited operating zone \( k \) of unit \( j \), respectively. \( \text{NPZ}_j \) is the number of prohibited operating zones of unit \( j \).

2.3. Constraint Handling Mechanism

2.3.1. Inequality Constraint Handling. The inequality constraints (4) are easy to meet, because the optimization variables are directly set within their ranges in the initialization process, and any variable beyond their bound can be limited to the bound which it exceeds for the proposed algorithm. For the inequality constraints (8), a common penalty function approach is used to handle it. If the output power of the generating unit \( j \) lies in the POZs, the new cost function of the ELD problems with the POZs is expressed by

\[ f_c = \sum_{j=1}^{N_G} F_j \left( P_j \right) + \delta \times \min \left( P_j - P_{j,k-1}, P_{j,1} - P_j \right) \]

(9)

where \( \delta \) is a penalty factor. \( \min \left( P_j - P_{j,k-1}, P_{j,1} - P_j \right) \) denotes constraint violations when the unit \( j \) operates in the POZs.

2.3.2. Equality Constraint Handling for the ELD Problems without Transmission Losses. Contrary to inequality constraints, the equality constraints are hard to satisfy. The penalty function method can be also used to help intelligent algorithms to find the optimization solutions, but the
quality of solutions cannot be ensured. Aim at the equality constraints, scholars proposed diverse repair methods for the ELD problems with or without transmission losses [9, 10, 13]. Due to the difference between the ELD problems without transmission losses and the ELD problems with transmission losses, two proposed repair methods will be, respectively, introduced. A repair method of this paper for the ELD problems without transmission losses is first introduced in this section.

To obtain more promising feasible solutions, a repair method and penalty function method are used to together cope with the equality constraint, and the equality constraint handling process is clearly shown in Algorithm 1.

\( \Delta j \) represents the adjustment limit of the component \( j \). \( |w| \) represents the equality constraint violation. sign is called sign function. When \( w \) is positive, the \( \text{sign}(w) \) value is equal to 1. On the contrary, when \( w \) is negative, the \( \text{sign}(w) \) value is equal to -1. The repair method will select a random component \( x^i_j \) from the individual vector \( x_i = [x^i_1, x^i_2, \cdots, x^i_N] \), \( i = 1, 2, \cdots, NP \), and let \( J = \{1, 2, \cdots, N_D \} \). If the selected component \( x^i_j \) is equal to the upper bound \( P_j^{\text{max}} \) (or lower bound \( P_j^{\text{min}} \) ) when the total generating power output is smaller (larger) than demand, this component will be removed from the set \( J \). Then a new component will be randomly selected from the set \( J \) that has excluded the previous component(s). This process will be repeated unless a component that is not equal to its upper (or lower) bound is found. The way of selecting a component can avoid some ineffective operations which do not repair those infeasible solutions. The smaller one in \( |w| \) and \( \Delta j \) is selected to adjust the output power of the unit, which is to avoid the violation of inequality constraints after executing the repair process. Similar to the reported literature [10], a penalty method is still used to further handle equality constraint. Furthermore, a sign flag is set to judge whether the equality constraint has been completely met after the repair operation. If the equality constraint violations are surely equal to zero, the penalty method will not be used to further handle the equality constraint. On the contrary, if the equality constraint violations are not equal to zero, the penalty method will be used to further handle the remaining constraint violations. The inequality constraints related to the POZs are also handled by the penalty method. Because the environment of all experiments is based on MATLAB, MATLAB precision has a certain effect on the optimization results. The advantage of the above setting flag is that it can decrease the effects of MATLAB precision on the optimization results as much as possible. If another programming language is used to implement the proposed mechanism, the flag variable can be also used to eliminate the error due to the other operating environments precision. Thus, the flag variable cannot be ignored when other programming languages are used.

2.3.3. Equality Constraint Handling for the ELD Problems with Transmission Losses. Qin and Cheng et al. [9] presented a repair method that can reduce the equality constraint violations until these violations are within the error tolerance \( \varepsilon \) (\( \varepsilon = 0.001 \)), which means that the repaired solutions are considered to be the feasible ones. A component \( j \) is randomly selected from the set \( J \). \( x^i_j \), \( J \), \( P_j \), \( |w| \), and sign represent the concept same as the aforementioned repair method. rand represents a randomly generated number in the interval \([0,1]\). The method also compares the \( |w| \) value with the \( \Delta j \) value to select a smaller one that is used to update solutions. If the \( |w| \) value is larger (smaller) than the \( \Delta j \) value, \( x^i_j \) will be subtracted to \( \text{sign}(w) \times \Delta j \) (or \( w \)). Then the transmission losses and equality constraint violations are recalculated. This above procedure is repeated until the \( |w| \) value is smaller than the error tolerance \( \varepsilon \). The proposed repair method is similar to the repair method of Qin and Cheng et al. [9]. The only difference is that \( \Delta j \) is calculated based on the adjustment limit on the component \( j \), and the repair method for the ELD problems with transmission losses is shown in Algorithm 2. This modification contributes to reduce more constraint violations and obtains the better solutions satisfying the tolerance error compared to the original mechanism. It will be indicated from Section 4.8 that the modification is effective.

3. Differential Evolution Algorithm and Its Improved Version

3.1. Differential Evolution Algorithm. Differential evolution (DE) algorithm, presented by Storn and Price [22], is a very competitive intelligent algorithm. It has simple structure and powerful capacity of searching solution space. Therefore, DE has been frequently applied to many fields for solving those real-word problems, such as optimal reactive power dispatch [23], parameter estimation problems in computational systems biology [24], iris recognition [25], and constraint optimization [26]. In this section, the traditional DE algorithm and its improved version will be introduced in detail.

The operation steps of DE are initialization, mutation, crossover and selection, respectively. In addition, a few parameters also play a key role on the DE's performance. These steps and parameters are explained as follows.

3.1.1. Initialization. The goal of this step is to generate \( NP \) candidate solutions consisting of a population. All candidate solutions are randomly generated within the lower and upper bounds of searching space. For example, the \( j \)th \((j = 1, 2, \cdots, N_D)\) component of the \( i \)th \((i = 1, 2, \cdots, NP)\) candidate solution is initialized by

\[
\begin{align*}
x^i_j(0) & = L^j + \text{rand} \times (U^j - L^j)
\end{align*}
\]

where \( L^j \) and \( U^j \) denote the lower and upper bounds of the \( j \)th component, respectively. \( x^j(0) \) denotes the \( j \)th variable of the \( i \)th individual at the initial generation.

3.1.2. Mutation Operation. In this step, Mutant vector \( v_i(t) \) is generated based on several randomly selected individuals for each parent individual vector \( x_i(t) \) in the current generation \( t \). DE/rand/1 is the most common and popular mutation strategy to generate mutant vector according to the following
Process: the equality constraint handling for the ELD problems without transmission losses

1. Initialize a set $J = \{1, 2, \cdots, N_G\}$; Set $w = \sum_{j=1}^{N_G} P_j - P_D$;
2. If $w < 0$
3. Randomly select a component $j$ from the set $J$;
4. While $x_j = P_j^{\text{max}}$
5. Remove $j$ from $J$, and let the new set be $J'$. Then randomly select a new component $j'$ from $J'$, and $j \leftarrow j'$, $J \leftarrow J'$;
6. End While
7. Calculate the adjustment limit of the component $j$, namely $\Delta_j = P_j^{\text{max}} - x_j$;
8. Else If $w > 0$
9. Randomly select a component $j$ from the set $J$;
10. While $x_j = P_j^{\text{min}}$
11. Remove $j$ from $J$, and let the new set be $J'$. Then randomly select a new component $j'$ from $J'$, and $j \leftarrow j'$, $J \leftarrow J'$;
12. End While
13. Calculate the adjustment limit of the component $j$, namely $\Delta_j = x_j - P_j^{\text{min}}$;
14. End If
15. Set $\text{flag} = 0$;
16. If $|w| > \Delta_j$
17. $x_j = x_j - \text{sign}(w) \times \Delta_j$;
18. $\text{flag} = 1$;
19. Else If $|w| < \Delta_j$
20. $x_j = x_j - w$;
21. End If
22. If $\text{flag} = 1$
23. A penalty function method is further used to handle equality constraint. The new objective function is represented as the formula $f_c = \sum_{j=1}^{N_G} F_j(P_j) + \delta \times |w|$
24. Else
25. The equality constraint is completely met. At this time, the objective function is represented as the formula $f_c = \sum_{j=1}^{N_G} F_j(P_j)$
26. End If

Algorithm 1: The pseudo code of the equality constraint handling process for the ELD problems without transmission losses.

Process: the equality constraint handling for the ELD problems with transmission losses

1. Initialize a set $J = \{1, 2, \cdots, N_G\}$; Set $w = \sum_{j=1}^{N_G} P_j - P_D - P_L$ and the error tolerance $\epsilon$;
2. While $|w| > \epsilon$
3. Randomly select a component $j$ from the set $J$;
4. If $w < 0$
5. $\Delta_j = P_j^{\text{max}} - x_j$;
6. Else If $w > 0$
7. $\Delta_j = x_j - P_j^{\text{min}}$;
8. End If
9. If $|w| > \Delta_j$
10. $x_j = x_j - \text{sign}(w) \times \Delta_j$;
11. Else If $|w| < \Delta_j$
12. $x_j = x_j - w$;
13. End If
14. Recalculate the transmission losses $P_L$ and equality constraint violations $|w|$;
15. End While

Algorithm 2: The pseudo code of the equality constraint handling process for the ELD problems with transmission losses.
vector \( \mathbf{v}_i(t) = x_{i3}(t) + F \times (x_{i1}(t) - x_{i2}(t)) \) \]

where \( i, i_1, i_2, \) and \( i_3 \) are three randomly selected integers in \([1, NP]\), and \( i_1 \neq i_2 \neq i_3 \neq i \). The control parameter \( F \) represents the scale factor that is used to scale the difference vector \( (x_{i1}(t) - x_{i2}(t)) \). The scale factor \( F \) is a user-defined parameter \((F > 0)\) [27] and usually set in the interval \((0, 1)\).

Except for this mutation strategy \( \text{DE/rand/1} \), another five popular mutation strategies are listed below [28]:

- **DE/best/1**
  \[
  v_i(t) = x_{\text{best}}(t) + F \times (x_{i1}(t) - x_{i2}(t))
  \]
  \( i_1 \neq i_2 \neq \text{best}, \ i_1 \neq i_2 \neq i \)

- **DE/rand/2**
  \[
  v_i(t) = x_{i1}(t) + F \times (x_{i2}(t) - x_{i3}(t)) + F \times (x_{i4}(t) - x_{i5}(t))
  \]
  \( i_1 \neq i_2 \neq i_3 \neq i_4 \neq i_5 \neq i \)

- **DE/best/2**
  \[
  v_i(t) = x_{\text{best}}(t) + F \times (x_{i1}(t) - x_{i2}(t))
  \]
  \( i_1 \neq i_2 \neq \text{best}, \ i_1 \neq i_2 \neq i \)

- **DE/current-to-best/1**
  \[
  v_i(t) = x_i(t) + F \times (x_{\text{best}}(t) - x_i(t)) + F \times (x_{i1}(t) - x_{i2}(t))
  \]
  \( i_1 \neq i_2 \neq \text{best}, \ i_1 \neq i_2 \neq i \)

- **DE/rand-to-best/1**
  \[
  v_i(t) = x_{i1}(t) + F \times (x_{\text{best}}(t) - x_{i1}(t)) + F \times (x_{i2}(t) - x_{i3}(t))
  \]
  \( i_1 \neq i_2 \neq i_3 \neq \text{best}, \ i_1 \neq i_2 \neq i_3 \neq i \)

where \( x_{\text{best}}(t) \) represents the best individual vector at generation \( t \).

### 3.1.3. Crossover Operation

Crossover has two types of strategies including binary crossover and exponential crossover. In this work, the binary crossover is adopted to obtain the trail vector \( u_i(t) = (u_{i1}(t), u_{i2}(t), \ldots, u_{iNP}(t)) \) whose components are got from those of both parent vector \( x_i(t) \) and mutant vector \( v_i(t) \) based on the crossover rate. This operation is expressed by

\[
u_i^j(t) = \begin{cases} 
  v_i^j(t), & \text{If rand} < CR \text{ or } j = j_{N_{ai}} \\
  x_i^j(t), & \text{Otherwise}
\end{cases}
\]

where another important control parameter \( CR \) stands for the crossover rate which determines the probability of selecting components of mutant vector \( v_i(t) (i = 1, 2, \cdots, NP) \). \( CR \) is a user-defined parameter set in the interval \([0, 1]\). \( j_{N_{ai}} \) is a randomly generated parameter between \( 1 \) and \( N_i \).

### 3.1.4. Selection Operation

The behavior of this operation is similar to a greedy strategy, and it only reserves better individuals into the next generation. For \( \text{DE} \), if the updated trail vector \( u_i(t) \) is better than parent vector \( x_i(t) \), the former will replace latter into the next generation. Otherwise, the corresponding offspring individual vector is the same as the parent individual vector, which means that the updating of the individual is not successful. For a minimization problem, the offspring individual \( x_i(t + 1) (i = 1, 2, \cdots, NP) \) is generated by

\[
x_i(t + 1) = \begin{cases} 
  u_i(t), & \text{If } f(u_i(t)) \leq f(x_i(t)) \\
  x_i(t), & \text{Otherwise}
\end{cases}
\]

where \( f(\cdot) \) stands for the fitness function, which is to calculate the fitness value of an individual. In this problem, the smaller the fitness value of individual is, the better the performance of individual is.

### 3.2. A Phase-Based Adaptive Differential Evolution Algorithm

A phase-based adaptive differential evolution algorithm (PADE) is proposed to enhance the \( \text{DE} \)’s ability in finding the global best solution. PADE and \( \text{DE} \) are different in three aspects.

#### 3.2.1. An Archive of Storing Successful Individuals

In the \( \text{DE} \)’s mutation strategy, the parent individual is usually randomly selected from the current population. This parent-selecting way is beneficial for maintaining the population diversity, but it may not jump out of local optima when stagnation is happening to \( \text{DE} \). Based on this issue, Guo and Yang et al. [29] established an archive of recently updated individuals to improve the parent-selecting way by utilizing the successful individuals. If the current parent individual fails to update beyond the stagnation tolerance on the number of consecutive unsuccessfully updates, the parent individuals of mutation strategy will be randomly selected from the archive instead of the current population. When the archive exceeds its size, the oldest individual is removed. On the contrary, Tanabe and Fukunaga [30] adopted an external archive that stores the failed solutions at each generation to maintain the population diversity. Inspired by the study, we also set up an archive \( S \) that stores the recently updated individuals. Different from the archive of [29, 30], the randomly selected individuals from the archive are used for the base vector of mutation strategy in the current generation. Then the recently updated individuals are stored to \( S \) that has been set to empty after each generation. In other words, \( S \) only stores the recently updated individuals. As long as the recently updated individual exists, the base vector of each generation is randomly chosen from the archive in the early phase. In addition, if there are not updated individuals in a generation,
the base vector of mutation strategy will be randomly selected from the current population. The new parent-selecting way has a greater chance to generate successful individuals, and maintains the population diversity. For the initialization of $S$, it is first initiated to be a copy of the initial population. Additionally, in case the initial population contains infeasible solutions, they will be also used for the mutation operation in the early evolutionary phase, because those infeasible solutions are gradually evolved to feasible ones as the evolution progresses.

3.2.2. A Phase-Based Mutation Operation. In DE, mutation is an important step for obtaining the mutant individuals in the area around the base individual. In this area, the base individual locates in the center, the difference vector acts as the movement direction and step, and the scale factor scales the movement step. In fact, there are many different types of the base individuals and difference vectors, which consist of several mutation strategies of different characteristics, such as $DE/rand/1$, $DE/rand/2$, $DE/best/1$, $DE/best/2$, and $DE/current-to-best/1$. According to [31], $DE/rand/1$ and $DE/rand/2$ are very helpful to maintain the diversity of population but they perform poor in convergence rate and are easy to gradually approach the local optima. On the contrary, it is known from [31] that $DE/best/1$ and $DE/best/2$ are beneficial to accelerate population convergence, because the base individual is the global best individual with good information, which can guide other individuals to approach it. However, it is obvious that they are very easy to fall into the local optima. Specially, it can lead to premature convergence when they are used in the early phase of evolution. $DE/current-to-best/1$ [32] employs the current individual as the origin individual and the best individual as the guiding individual, which constructs a mixed base individual. $DE/current-to-best/1$ is called composite mutation strategies, which has better diversity than $DE/best/1$, but its global exploration is poor. In the entire evolutionary process, global exploration and local exploitation are two vital aspects, which should cooperate well with each other to find the promising solutions. First, the algorithm should widely search in the global area and find some local areas with promising individuals in the search space. Then, the algorithm should focus on local exploitation in these local areas. It is to enhance the ability of finding the optimal solution as fast as possible. In DE, the population individuals spread over the searching space in the early phase of evolution. They can update themselves based on other individuals through global search but it can also cause premature convergence. In the late phase of evolution, most individuals gather in a relatively small area, and they implement local exploitation instead of global exploration.

Based on the above analysis, Miranda-Varela and Mezura-Montes [33] relied on the number of feasible solutions to select one between $DE/rand/1$ and $DE/best/1$ to balance the exploration and exploitation ability. Qin and Huang et al. [31] constructed a mutation strategy candidate pool that contains several mutation strategies with diverse characteristics. In the evolution process, one strategy will be selected from the candidate pool according to a probability learned from its previous experience of generating promising solutions for each parent individual. The more successfully one strategy behaved in previous generations, the more probably it will be selected in the current generation. Different from the above studies, to match different phases of evolution, $DE/srand/1$ and $DE/best/1$ are respectively used to execute the mutation operation in the early and late phase of evolution in PADE. For $DE/srand/1$, its base vector is randomly selected from the archive $S$, and its difference vector is the same as $DE/rand/1$. Therefore, $DE/srand/1$ is more easy to find the successful individuals than $DE/rand/1$. Now, the key task is how to select a generation index threshold between 1 and $T$ to divide the entire evolution process into the early and late phase. Many scholars tend to set a probability rule based on the number of iterations [34, 35] or the fitness value of random individual [36] to determine whether the current evolution lies in the early phase or late phase of iteration. Different from their methods, Tang and Dong et al. [37] introduced the success ratio $SR$ and an auxiliary generation index threshold $G_t$ to help select the generation index threshold $g_t$. The threshold $g_t$ is set based on the recently updated individuals, which is more heuristic for enhancing the performance of DE than the above two methods. In view of the advantage of the success ratio, this paper employs it to divide the evolution process into the early and late phase. The success ratio of each generation $t$ is defined by:

$$SR_t = \frac{NS_t}{NP}$$  \hspace{1cm} (19)

where $NS_t$ is the number of the updated individuals at generation $t$, $SR_t$ stands for the success ratio at generation $t$. In general, $SR_t$ is a relative large value in the early phase of evolution. As the generation advances, $SR_t$ is prone to decrease continually, so it can be used to determine whether the late phase begins. When $SR_t$ is smaller than a threshold of success ratio $SR_K$ for a given number of consecutive generations $N_{succ}^{max}$, the late phase of the algorithm begins. If $SR_t$ is larger than $SR_K$ at generation $t$, the current number of consecutive generations is denoted as $N_{succ} = 0$. Otherwise, $N_{succ} = N_{succ} + 1$. If $N_{succ}$ reaches $N_{succ}^{max}$, $N_{succ}$ keeps the fixed value $N_{succ}^{max}$ until the evolution ends. That means that PADE only uses the mutation strategy $DE/best/1$ in the late phase. For most problems, $SR_t$ can decrease to $SR_K$ for consecutive generations $N_{succ}^{max}$ in the early phase of evolution. However, it should be noted that $SR_t$ cannot decrease to $SR_K$ for consecutive generations $N_{succ}^{max}$ in the early or entire evolution phase for a few problems. Therefore, under these circumstances, a threshold $t_{force}$ is forced to set to divide the process into two phases. The threshold $t'$ determined by success ratio is given by:

$$t' = \{ t \mid SR_t < SR_K, N_{succ} = N_{succ}^{max}, \forall t \in [1, T] \}$$  \hspace{1cm} (20)

In order to comprehensively describe the phase-based mutation operation producing the mutant vector, the following formula is given by
where the index of successful individual \( s_{rand} \) is randomly integer between 1 and the size of the archive \( S \), and \( s_{rand} \neq i1 \neq i2 \neq i \).

To visualize this phase-based mutation operation from an overall perspective, its schematic diagram is presented as follows (see Figure 1).

In Figure 1, the entire evolution phase is divided into the early and late phase. There are two situations on determining the threshold, which are represented as S1 and S2, respectively. Specifically, above the 0 to \( T \) axis, the way of dividing the entire evolution phase is represented as S1. Below the 0 to \( T \) axis, the way of dividing the entire evolution phase is represented as S2. The threshold \( t' \) can lie in \([ t_{\text{force}} , T ]\), or it does not exist in the \([0 , T] \), where the \( t_{\text{force}} \) is used as the threshold at this time just as S2.

### 3.2.3. Adaptive Control Parameters

The two control parameters, scale factor \( F \) and crossover rate \( CR \), play a significant role on the performance of DE. Their values are traditionally fixed as constants to participate in evolution. A suitable \( F \) and \( CR \) are related to a specific problem, and all individuals share their fixed control parameters in the evolution process for the original DE. The result is that the robustness of the algorithm is poor, and a lot of tedious optimization trials must be done to find a suitable \( F \) and \( CR \) for practical problems. To enhance the robustness of the algorithm and avoid tedious optimization trials, many researchers adopted different self-adaptive strategies to make control parameters self-adaptively change with individual evolution [36, 38]. Some of them randomly take a value within the ranges of parameters owned by the individual when the individual doesn’t survive into the next generation. The other do not have the guidance rule at all, and the control parameters are changed based on a probability in the search process. In addition, there are some works [39, 40] only for this type of problems, where self-adaptive approaches have proven to be competent in different problems, for example, constraint optimization problem and biobjective tooth profile spur gear optimization problem. Qin and Huang et al. [31] proposed that the scale factor values are randomly sampled from a normal distribution and applied to each parent individual in the current population, which is helpful for the global and local of population. An adaptation scheme is used to generate the crossover rate and is based on the previous experiences of generating promising individuals. However, these self-adaptive strategies cannot well guide the population evolution for this type of ELD problems. It is clearly different that the two adaptive parameters with the learning ability are used to evolve with individuals in this paper. Because the best individuals usually own best control parameters in the improved population, control parameters of other individuals are required to learn from those of the best individuals, which is beneficial to find the global best individual. Same as other self-adaptive strategies, two control parameters are first encoded as shown in Figure 2.

Every individual vector \( x_i(t) \) owns two control parameters \( F_i(t) \) and \( CR_i(t) \). In the initial process, \( F_i(t) \) and \( CR_i(t) \) are randomly generated in a given range \([ F_{\text{min}}, F_{\text{max}}] \) and \([ CR_{\text{min}}, CR_{\text{max}}] \), respectively. Since the population is not updated during the first generation, the two control parameters of the failed to update individuals are regenerated, which is the same as their initialization process. In the next evolution, if the current population has been improved in the \( t \)th generation (marked as \( \text{fail} = 0 \)), two control parameters of the failed to update individuals will learn from those of the best individual. Otherwise (marked as \( \text{fail} = 1 \)), their control parameters are randomly generated in their ranges. If the current individual is successfully improved at generation \( t \), its corresponding parameters keep the previous value. That means that control parameters of the current individual are helpful to improve individual. Based on the above explanation, adaptive parameters with learning ability are described by

\[
F_i(t + 1) = \begin{cases} 
F_i(t) + \text{rand} \times (F_{\text{best}}(t) - F_i(t)), & \text{if } f(u_i(t)) > f(x_i(t)) \text{ and } t > 1 \text{ and } \text{fail} = 0 \\
F_{\text{min}} + \text{rand} \times (F_{\text{max}} - F_{\text{min}}), & \text{if } f(u_i(t)) > f(x_i(t)) \text{ and } (t \leq 1 \text{ or } \text{fail} = 1) \\
F_i(t), & \text{Otherwise}
\end{cases}
\]

\[
CR_i(t + 1) = \begin{cases} 
CR_i(t) + \text{rand} \times (CR_{\text{best}}(t) - CR_i(t)), & \text{if } f(u_i(t)) > f(x_i(t)) \text{ and } t > 1 \text{ and } \text{fail} = 0 \\
CR_{\text{min}} + \text{rand} \times (CR_{\text{max}} - CR_{\text{min}}), & \text{if } f(u_i(t)) > f(x_i(t)) \text{ and } (t \leq 1 \text{ or } \text{fail} = 1) \\
CR_i(t), & \text{Otherwise}
\end{cases}
\]

where \( F_i(t) \) and \( CR_i(t) \) are the scale factor and crossover rate of the current \( i \)th individual at generation \( t \). \( F_i(t + 1) \) and \( CR_i(t + 1) \) are the scale factor and crossover rate of the \( i \)th individual at generation \( t + 1 \). \( F_{\text{best}}(t) \) and \( CR_{\text{best}}(t) \) are the best control parameters owned by the best individual in the current population, \( F_{\text{min}} \) and \( F_{\text{max}} \) denote the minimum and maximum scale factor, respectively. \( CR_{\text{min}} \) and \( CR_{\text{max}} \) denote the minimum and maximum crossover rate, respectively. When control parameters of the current individual learn from the best control parameters, its learning step is a
The entire evolution phase

Figure 1: Schematic diagram of dividing the evolution phase.

The early phase

The late phase

S1

S2

Fisher

The early phase

The late phase

\[ x_1(t) = (x_1^1(t), x_1^2(t), \ldots, x_1^{N_c}(t)) \]

\[ x_2(t) = (x_2^1(t), x_2^2(t), \ldots, x_2^{N_c}(t)) \]

\[ x_{NP}(t) = (x_{NP}^1(t), x_{NP}^2(t), \ldots, x_{NP}^{N_c}(t)) \]

Figure 2: Description of encoding control parameters.

3.3. The Pseudo Code of the Proposed Algorithm. To fully demonstrate the computational steps of PADE, the algorithmic pseudo code is presented in Algorithm 3.

4. Simulation Results and Analysis

4.1. Experimental Cases and Parameter Settings. In this section, there are six ELD cases selected to verify the superiority of PADE compared to the other five DE variants and methods from the literature, and the parameters of experiment are set in advance. To be more specific, these cases include (1) the 6-unit system \((PD = 1263\text{MW})\) [41]; (2) the 6-unit system with transmission losses \((PD = 1200\text{MW})\) [3]; (3) the 10-unit system with VPE and transmission losses \((PD = 2000\text{MW})\) [14]; (4) the 14-unit system with transmission losses \((PD = 950\text{MW})\) [42]; (5) the 40-unit system with VPE \((PD = 1050\text{MW})\) [2]; (6) the 40-unit system with VPE and POZs \((PD = 1050\text{MW})\) [4]. Cases 2-4 consider the transmission losses of power system, and case 3 also considers VPE. Meanwhile, the cases with VPE also include the last two cases, and the last case takes the POZs into consideration. The detailed information of these cases can be obtained from the corresponding literature mentioned above [2–4, 14, 41, 42], and the prohibited operation zones are taken from Table 12 of [4].

The PADE algorithm is compared with other five state-of-the-art DE variants including differential evolution algorithm with self-adapting control parameters (jDE) [43], differential evolution algorithm with controlled search direction and self-adapting control parameters (jdDE) [44], modified differential evolution algorithm (mDE) [45], differential evolution with composite trial vector generation strategies and control parameters (CODE) [46], and improving differential evolution with a successful-parent selecting framework (SPS-DE) [29]. The parameters of the six DE variants are set as follows: for jDE, \(F_1 = 0.1, F_u = 0.9\), and \(\tau_1 = \tau_2 = 0.1\). The scale factor and crossover rate are, respectively, set to 0.6 and 0.3 in the initial generation, which cannot be obtained from [43]. For jdDE, the values of \(F_1, F_u, \tau_1, \tau_2\), and \(\tau_1\) and \(\tau_2\) are the same as the above corresponding values in jDE. \(CR_j = 0.1\) and \(CR_u = 1.0\). For mDE, the fixed crossover rate is taken as \(CR = 0.8\). For CODE, except for the maximum function evaluation number \(MAXFE\) [47, 48], the other parameters are obtained by [46]. For SPS-DE, the stagnation tolerance \(Q\) is set to 32, and the SPS framework is incorporated with the standard DE with the strategy \(DE/rand/1/bin\), scale factor \(F = 0.6\) and crossover rate \(CR = 0.3\). For the proposed algorithm, \(F_{\min} = 0.1, F_{\max} = 0.9\), \(SR_K = 0.1\), \(t_{\text{force}} = 90\% \times T\), and \(N_{\text{succ}}^{\text{max}} = \left[3e^{N_c}\right]\). \(CR_{\text{min}} = 0.1\) and \(CR_{\text{max}} = 0.3\) are set for the last two cases, and the crossover rate ranges between 0.3 and 0.6 for the remaining cases. It should be noted that these ELD cases can find the most promising solutions by our parameter settings. For all simulation experiments, the population size is set to 40, and they have the same \(MAXFE\) when all six DE variants optimize the same ELD case, which make the results comparison fair. In addition, the maximum generation number \(T\) of jDE, jdDE, mDE, SPS-DE, and PADE are same for a ELD case. The maximum generation number \(T\) is equal to 1000, 300, 300, 300, 4000, and 4000 for cases 1-6, respectively. The \(MAXFE\) is equal to 40000, 12000, 12000, 12000, 160000, and 160000 for cases 1-6, respectively. When the penalty method is used to handle constraints, the penalty factor \(\delta\) is set to \(10^8\). Additionally, Matlab8.3 simulation software is used to execute all experiments under the environment of an Intel (R) Core (TM) i5-2450M CPU @
Algorithm: phase-based adaptive differential evolution

(1) Initialize a population \( \text{pop}_0 \);

(2) Set population size \( NP \) and the dimension of the individual \( N_G \); the maximum generation number \( T \); the maximum function evaluation number \( \text{MAX}_{FE} \); the initial individual control parameters and their bounds; \( N_{\text{max}}; SR_G; S = \text{pop}_0; \)

\( t = 0; N_{\text{succ}} = 0; \) the cases including the 6-unit system \( (PD = 1263\text{MW}) \) [41], the 6-unit system with transmission losses \( (PD = 1200\text{MW}) \) [3], the 10-unit system with VPE and transmission losses \( (PD = 1200\text{MW}) \) [41], the 14-unit system with transmission losses \( (PD = 950\text{MW}) \) [42], the 40-unit system with VPE \( (PD = 10500\text{MW}) \) [2], the 40-unit system with VPE and POZs \( (PD = 10500\text{MW}) \) [4];

(3) While \( t \) doesn’t reach \( T \)

(4) For \( t = 1 \) to \( T \)

(5) Set \( N_S = 0; \) Obtain the global best solution \( x_{\text{best}}(t) \);

(6) For \( i = 1 \) to \( NP \)

(7) If \( N_{\text{succ}} < N_{\text{max succ}} \)

(8) If \( t < 90\% \times T \)

(9) If \( S \) is empty

(10) Randomly generate three integers \( i_1, i_2, \) and \( i_3 \) in the range \([1, NP]\), and \( i_1 \neq i_2 \neq i_3 \neq i \);

(11) \( v_i(t) = x_{i_3}(t) + F_i(t) \times (x_{i_1}(t) - x_{i_2}(t)); \)

(12) Else

(13) Randomly generate two integers \( i_1 \) and \( i_2 \) in the range \([1, NP]\), and randomly generate a integer \( \text{rand} \) between 1 and the size of the archive \( S \), and \( i_1 \neq i_2 \neq \text{rand} \neq i \);

(14) \( v_i(t) = x_{\text{rand}}(t) + F_i(t) \times (x_{i_1}(t) - x_{i_2}(t)); \)

(15) End If

(16) Else

(17) Randomly generate two integers \( i_1 \) and \( i_2 \) in the range \([1, NP]\), and \( i_1 \neq i_2 \neq \text{best} \);

(18) \( v_i(t) = x_{\text{best}}(t) + F_i(t) \times (x_{i_1}(t) - x_{i_2}(t)); \)

(19) End If

(20) Else

(21) Randomly generate two integers \( i_1 \) and \( i_2 \) in the range \([1, NP]\), and \( i_1 \neq i_2 \neq \text{best} \);

(22) \( v_i(t) = x_{\text{best}}(t) + F_i(t) \times (x_{i_1}(t) - x_{i_2}(t)); \)

(23) End If

(24) Randomly generate a integer \( j_{N_G} \) in the range \([1, N_G]\);

(25) For \( j = 1 \) to \( N_G \)

(26) If \( \text{rand} < CR_i(t) \) or \( j = j_{N_G} \)

(27) \( u_i(t) = v_i(t); \)

(28) Else

(29) \( u_i(t) = x_i(t); \)

(30) End If

(31) End For

(32) If \( f(u_i(t)) > f(x_i(t)) \)

(33) \( x_i(t + 1) = u_i(t); \)

(34) If \( t > 1 \)

(35) If \( \text{fail} = 0 \)

(36) \( F_i(t + 1) = F_i(t) + \text{rand} \times (F_{\text{best}}(t) - F_i(t)); \)

(37) \( CR_i(t + 1) = CR_i(t) + \text{rand} \times (CR_{\text{best}}(t) - CR_i(t)); \)

(38) Else

(39) \( F_i(t + 1) = F_{\text{min}} + \text{rand} \times (F_{\text{max}} - F_{\text{min}}); \)

(40) \( CR_i(t + 1) = CR_{\text{min}} + \text{rand} \times (CR_{\text{max}} - CR_{\text{min}}); \)

(41) End If

(42) Else

(43) \( F_i(t + 1) = F_{\text{min}} + \text{rand} \times (F_{\text{max}} - F_{\text{min}}); \)

(44) \( CR_i(t + 1) = CR_{\text{min}} + \text{rand} \times (CR_{\text{max}} - CR_{\text{min}}); \)

(45) End If

(46) Else

(47) \( x_i(t + 1) = u_i(t); \) \( N_S = N_S + 1; \)

Algorithm 3: Continued.
2.50GHZ. The simulation results of 30 independent runs can be obtained in Table 1.

4.2. Comparison among Six DE Variants for Six ELD Problems.
In order to improve the quality of solutions, an improved DE and one of the proposed repair methods are combined to find the best feasible solution for a specific case. In this section, all six DE variants are combined with the corresponding one of the proposed two repair methods that are different for the ELD problems without transmission losses and the ELD problems with transmission losses. Thus the performance of the six DEs is only compared in this section while the quality of repair methods will be discussed in Section 4.8.

In Table 1, several criteria are adopted to compare the performance of six improved DEs, including \( \text{Cost}_{\text{min}} \), \( \text{Cost}_{\text{max}} \), \( \text{Cost}_{\text{mean}} \), \( \text{Cost}_{\text{median}} \) and \( \text{Cost}_{\text{std}} \), which stand for the minimal, maximal, mean, median and standard deviation values of 30 test results, respectively. For case 1, all six DE variants are able to exactly find the global best solution, and all six DE variants have low \( \text{Cost}_{\text{std}} \) values, indicating that their stability is strong. For case 2, the minimal cost obtained by the first three comparison algorithms are the same as that of PADE, but the performance of PADE is better than that of the three algorithms according to the \( \text{Cost}_{\text{max}}, \text{Cost}_{\text{std}} \). The \( \text{Cost}_{\text{min}} \) values of the remaining two comparison algorithms are slightly larger than that of PADE. In terms of the \( \text{Cost}_{\text{mean}} \), the results of PADE and jDE are both equal to the minimal cost function value, and they are smaller than those of the other four algorithms. The \( \text{Cost}_{\text{median}} \) values of PADE, jDE, and jDE are also slightly equal to the minimal cost function value, which are slightly smaller than those of mDE, CODE, and SPS-DE. Overall, the \( \text{Cost}_{\text{max}}, \text{Cost}_{\text{mean}}, \text{Cost}_{\text{median}} \) of PADE are equal to its \( \text{Cost}_{\text{min}} \) value, and its \( \text{Cost}_{\text{std}} \) value is close to 0. It means that PADE is easy to find the smallest objective function value in each run and has strong stability for this case. For case 3, both jDE and PADE can find the same lowest cost compared to the other four DEs. Furthermore, the \( \text{Cost}_{\text{max}}, \text{Cost}_{\text{mean}}, \text{Cost}_{\text{median}}, \text{Cost}_{\text{std}} \) values of PADE are smaller than those of jDE. Thus PADE is superior to jDE in stability and success rate. All the results obtained by jDE, mDE, CODE, and SPS-DE are worse than those of the other two competitors, and mDE performs the worst among all methods in terms of the \( \text{Cost}_{\text{std}} \), explaining that the stability of mDE is poor.

With regard to case 4, PADE can obtain the best \( \text{Cost}_{\text{min}}, \text{Cost}_{\text{mean}} \) and \( \text{Cost}_{\text{median}} \) compared to the other approaches. The \( \text{Cost}_{\text{min}} \) value of mDE is slightly larger than that of PADE, and it is acceptable. However, unfortunately, the \( \text{Cost}_{\text{max}}, \text{Cost}_{\text{mean}}, \text{Cost}_{\text{median}}, \text{Cost}_{\text{std}} \) values of mDE are far larger than those of another five approaches. Therefore, although mDE has the strong global search capability close to PADE, it performs poorly in stability. The \( \text{Cost}_{\text{max}} \) and \( \text{Cost}_{\text{std}} \) values of PADE are slightly larger than those of jDE, and they are smaller than those of the other four DEs. Therefore, the results of PADE are acceptable. jDE has the advantage over another four comparison algorithms in the average and median cost function values. On the contrary, jDE has poor performance according to the \( \text{Cost}_{\text{min}} \). In addition, CODE has the worst global search ability among all six algorithms, but its \( \text{Cost}_{\text{std}} \) value is acceptable. It means that CODE has a great chance to get trapped into a local optimum over 30 runs. For case 5, PADE has absolute advantage over another five algorithms, because it is able to find the best \( \text{Cost}_{\text{min}}, \text{Cost}_{\text{max}}, \text{Cost}_{\text{mean}}, \text{Cost}_{\text{median}}, \text{Cost}_{\text{std}} \), and these values are even smaller than those of another five algorithms. It suggests that PADE performs the best among all six DEs on solving the case 5. For case 6, although system considers POZs, PADE still has great advantage over five competitors. The \( \text{Cost}_{\text{min}}, \text{Cost}_{\text{mean}}, \text{Cost}_{\text{median}}, \text{Cost}_{\text{std}} \) values of PADE are better than those of another five competitors, which exhibits that the performance of PADE is very competitive. In fact, the ELD

\[
\begin{align*}
\text{(48)} & \quad \text{Store the updated individuals into } S \text{ at generation } t \text{ for the next generation, and } S \text{ only reserves the updated individuals in the current population;}
\text{(49)} & \quad \text{End If}
\text{(50)} & \quad \text{End For}
\text{(51)} & \quad \text{If } N_{\text{succ}} < N_{\text{max}}
\text{(52)} & \quad \text{If } \text{SR}_t < \text{SR}_K
\text{(53)} & \quad \quad N_{\text{succ}} = N_{\text{succ}} + 1;
\text{(54)} & \quad \text{Else}
\text{(55)} & \quad \quad N_{\text{succ}} = 0;
\text{(56)} & \quad \text{End If}
\text{(57)} & \quad \text{End If}
\text{(58)} & \quad \text{If the current population is improved}
\text{(59)} & \quad \quad \text{Update the global best solution;}
\text{(60)} & \quad \quad \text{fail} = 0;
\text{(61)} & \quad \text{Else}
\text{(62)} & \quad \quad \text{fail} = 1;
\text{(63)} & \quad \text{End If}
\text{(64)} & \quad \text{End For}
\text{(65)} & \quad \text{End While}
\end{align*}
\]

Algorithm 3: The pseudo code phase-based adaptive differential evolution.
problems with POZs are more complex than those without POZs. However, it is clearly seen that the Cost\(_{\text{min}}\), Cost\(_{\text{max}}\), Cost\(_{\text{mean}}\), and Cost\(_{\text{median}}\) obtained by PADE for case 6 are slightly smaller than, slightly larger than, or equal to, those of PADE for case 5. Therefore, the proposed algorithm still has strong search ability of solution space when the ELD problems become more complex. Additionally, the Cost\(_{\text{min}}\), Cost\(_{\text{mean}}\), and Cost\(_{\text{median}}\) obtained by jDE for case 6 are even smaller than those of it for case 5. It explains that jDE also has strong search ability for the complex ELD problems. Contrary to PADE and jDE, the results of jdDE and mDE on solving the case 6 are far larger than those of them on solving the case 5 except the Cost\(_{\text{std}}\) value of mDE. Thus the POZs of the ELD problems make it more difficult to find the feasible solutions for jDE and mDE. In addition, the results of CODE and SPS-DE are worse than those of PADE except the Cost\(_{\text{max}}\) value of SPS-DE for case 6. In short, jDE has good performance, but PADE performs the best among them.

In order to obtain the statistical significant difference when the proposed algorithm is compared to those comparison algorithms, Wilcoxon rank-sum test at the 0.05 significance level [51] is used in this section. The marks “+,” “=,” and “−” represent that the performance of PADE is better than, equal to, and worse than that of the corresponding comparison algorithms, respectively. It should be emphasized that Wilcoxon rank-sum test can only obtain whether the two algorithms have the difference. In other words, the better one cannot be still determined when the two algorithms have the

---

**Table 1: Comparison among six DEs for six ELD problems.**

| Problems | Algorithms | Cost\(_{\text{min}}\) | Cost\(_{\text{max}}\) | Cost\(_{\text{mean}}\) | Cost\(_{\text{median}}\) | Cost\(_{\text{std}}\) |
|----------|------------|----------------|----------------|----------------|----------------|----------------|
| Case 1   | jDE        | 15275.930392  | 15275.930392  | 15275.930392  | 15275.930392  | 5.500256e-12 = |
| Case 2   | jDE        | 63975.710808  | 63975.710809  | 63975.710809  | 63975.710808  | 8.430184e-08 = |
| Case 3   | jDE        | 111497.562764 | 111497.563832 | 111497.563072 | 111497.563001 | 2.525372e-04 = |
| Case 4   | jDE        | 4303.508000   | 4303.508294   | 4303.508114   | 4303.508106   | 7.29954e-05 = |
| Case 5   | jDE        | 121424.035506 | 121593.929338 | 121466.393369 | 121469.287345 | 2.447747e-05 |
| Case 6   | jDE        | 121420.895349 | 121848.603334 | 121420.895349 | 121420.895349 | 2.509872e-05 |

---


significant difference. If PADE and one of its competitors do not have significant difference, they are marked as “≈.” Based on the Wilcoxon rank-sum test characteristic, when PADE and any of the other five DEs have the significant difference, the criterion Cost\_min is used to determine whether PADE is statistically better than the other algorithm. The better one is marked as “+,” and the worse one is marked as “−.”

The results of Wilcoxon rank-sum test statistical analysis are listed in the last column of Table 1. For case 1, all five comparison groups do not have the significant difference, suggesting that the performance of comparison algorithms is equal to that of PADE. The conclusion can be also reflected from the corresponding Cost\_min, Cost\_max, Cost\_mean, Cost\_median, and Cost\_std values. For case 2, there is no difference between PADE and each of jDE, jdDE and mDE. Statistically, the performance of CODE and SPS-DE is worse than that of PADE. For case 3, PADE significantly outperforms jDE, mDE, CODE, and SPS-DE, and PADE beats them in comparison on this problem. For case 5, all competitors are completely beat by PADE. For case 6, although the problem becomes complicated, PADE is still the best algorithm compared to the other DE variants. In sum, PADE is statistically no worse than the other five algorithms in all cases. In addition, PADE significantly beats all competitors for cases 5-6. Therefore, PADE can efficiently solve the six ELD problems.

4.3. Convergence of Six DEs for The Fuel Cost Minimization

Under the condition that the proposed repair methods have been combined with six DEs to easily meet all the requirements of different constraints for the ELD problems, we now devote to study convergence of six DEs for the fuel cost minimization. In order to visualize the convergence characteristics of six DEs, in Figures 3(a)–3(f), we show the average convergence curves of six DEs for the fuel cost minimization of the selected six ELD problems over 30 runs, and these curves are based on the new objective function where the fitness function and penalty function are added for all ELD problems. The abscissa represents the function evaluations, and the ordinate represents the average fuel cost.

From Figures 3(a)–3(f), we can find that the convergence speed of PADE isn’t the fastest among all six DEs for all ELD cases, but the average curves of PADE can gradually become better than, or as good as those of the other five DEs in all cases as the evolution progresses. For the first case, the convergence speed of PADE ranks third among all six DEs. In Figure 3(b), mDE, CODE and SPS-DE do not converge to the feasible zone while the other three DEs can find the feasible solutions, and the average curve of PADE is the lowest among three visible curves. In Figure 3(c), the average curve of CODE is the highest among all six DEs, because it does not lie in the feasible range. On the contrary, PADE performs the best according to the average convergence curve. In Figure 3(d), it can be clearly seen that the average curve of PADE reaches a lower level compared to those of the other five DEs. For the first four figures, the feasible solutions are only contained, because their constraint violations descend rapidly. For cases 5-6, the average curves of all six DEs descend very fast at the beginning of evolution. To be more specific, some infeasible solutions contain in the figures for the last two cases. Their constraint violations cannot descend rapidly. Thus, the new objective function values are very large at the beginning of evolution. The declining speed of constraint violations can be seen from the upper bound of ordinate. If constraint violations exist, the penalty function value combining the penalty factor \((10^{30})\) multiplies and the constraint violations will be very large. However, they can gradually find the feasible solutions. Furthermore, the solutions obtained by PADE are better than those obtained by the other five DEs in complex ELD cases.

4.4. The Influence of the Introduced Archive

In this paper, the introduced archive \(S\) that stores the recently updated individuals is used to generate the promising feasible solutions with a higher probability in the early phase of evolution. To show the effect of the archive \(S\), PADE without archive is compared with PADE with archive, where the archive component of the algorithm is eliminated to test its contributions in improving the quality of solutions. The two PADE algorithms adopt the same proposed repair methods. Furthermore, three criteria are used to compare the performance of the two PADE versions, and they are \(Cost_{\text{mean}}\) and \(Cost_{\text{std}}\) of 30 test results and the total constraint violation \(V_c\) of the best solution in 30 test results, respectively. The \(V_c\) value contains the violations of all the equality and inequality constraints for a solution. The experimental parameters are the same as those of Section 4.2. The comparison results are listed in Table 2.

According to Table 2, the \(Cost_{\text{mean}}\) obtained by PADE with archive is the same as that of PADE without archive for cases 1-2, and their \(V_c\) values are equal to 0 for case 1. The \(Cost_{\text{std}}\) value of PADE with archive is slightly smaller than that of PADE without archive for the first two cases, meaning that the archive slightly enhances the stability in PADE. For cases 3-4, the \(Cost_{\text{mean}}\) and \(Cost_{\text{std}}\) values of PADE with archive are slightly smaller than those of PADE without archive, which indicates that PADE with archive performs more stable than PADE without archive. With regard to the last two cases, the archive improves the quality of solutions more obvious compared to the first four cases. To be more specific, the \(Cost_{\text{mean}}\) and \(Cost_{\text{std}}\) values of PADE with archive are far smaller than those of PADE without archive for cases 5-6. For cases 1-4, the performance of the two PADE versions is similar in the constraint violation. PADE with archive achieves a lower level than PADE without archive for cases 5-6 in the constraint violation, which suggests that the archive motivates PADE to find a more satisfactory solution. In all, PADE with archive performs more outstanding than PADE without archive.

4.5. The Influence of the PADE Parameters on the Solution Quality and Stability

In order to investigate the influence of success ratio \(SR_k\) on the solution quality and stability, the \(SR_k\) value is set to 0.1, 0.3, 0.5, 0.7, and 0.9, respectively. The other experimental parameters are the same as those of Section 4.2.
(a) The 6-unit system ($PD = 1263$MW)

(b) The 6-unit system with transmission losses ($PD = 1200$MW)

(c) The 10-unit system with VPE and transmission losses ($PD = 2000$MW)

(d) The 14-unit system with transmission losses ($PD = 950$MW)

(e) The 40-unit system with VPE ($PD = 10500$MW)

(f) The 40-unit system with VPE and POZs ($PD = 10500$MW)

Figure 3: Average convergence curves of six DEs for the fuel cost minimization of the selected six cases.
The optimization results of PADE over 30 runs are shown in Table 3.

Cost and Cost stand for the mean and standard deviation values of 30 test results, respectively. The Cost and Cost values are used to measure the solution quality and stability with respect to different SK values, respectively. It can be seen that all the Cost values are same with respect to different SK values for cases 1-2, meaning that the SK value does not affect the solution quality. For the remaining cases, the Cost values with respect to SK = 0.1 are better than those with respect to other SK values, indicating that SK = 0.1 is more suitable than other values in improving the solution quality. Additionally, the Cost values with respect to SK = 0.1 are the best among all the Cost values with respect to different SK values for four out of six cases, which suggests that PADE with SK = 0.1 has more stronger stability than PADE with other SK values. In sum, SK = 0.1 is suitable, since it is beneficial for the solution quality and stability.

To investigate the influence of their adaptive mechanism for two control parameters on the solution quality and stability, PADE with adaptive control parameters is compared with PADE with fixed control parameters by the criteria Cost and Cost. The fixed F and CR values are set to 0.6 and 0.3, respectively. The other experimental parameters are the same as those of Section 4.2. The comparison results are listed in Table 4.

According to Table 4, it is clearly seen from the terms Cost and Cost that PADE with adaptive control parameters is superior to PADE with fixed control parameters for all cases except case 1. For case 1, the solution quality and stability of PADE with adaptive control parameters and PADE with fixed control parameters are same. In all, the proposed adaptive control parameters have a significant effect in improving the performance of PADE for most cases.

4.6. Comparison between PADE with Phase-Based and Five Popular Mutation Operators. To test the effect of the proposed phase-based mutation compared to five popular mutation operators, PADE with phase-based mutation are compared with PADE with different mutation operators on the solution quality and stability. The other experimental parameters are the same as those of Section 4.2. The comparison results are presented in Table 5.

In Table 5, the Cost value of PADE with phase-based mutation is better than, or as good as that of PADE with other
mutation operators for the first two cases. For cases 3–4, the \( \text{Cost}_{\text{mean}} \) value of PADE with phase-based mutation is slightly larger than that of PADE with DE/best/1 and is lower than that of PADE with other mutation operators. It means that the solutions obtained by PADE with phase-based mutation are slightly worse than those of PADE with DE/best/1 but are better than those of PADE with other mutation operators. PADE with phase-based mutation outperforms PADE with another five mutation operators on the solution quality and stability for the last two complex ELD problems.

### 4.7. Comparison between PADE and the Other Methods Presented in the Literature

In this section, the best results of PADE are compared with those presented in the literature to evaluate the performance of different methods. Different from most literatures, we not only give the best cost function value (\( \text{Cost} \)), but also calculate and list the constraint violations of their best solutions listed in the paper. Here, the constraint violations are caused by power output limits, power balance constraints with or without transmission losses, or POZs for six studied ELD problems. The constraint violations of power output limits, power balance constraints with or without transmission losses, or POZs are determined to be acceptable and vice versa for cases 2–4. In addition, since the proposed repair method is easy to completely meet, so its error tolerance (\( \epsilon \)) is set to 0. However, because the actual ELD problems usually perform nonlinear characteristics especially for the ELD problems with transmission losses, the equality constraint, that is power balance constraint, can be slightly relaxed this restriction of the error tolerance. For the ELD problems with transmission losses, the equality constraint, \( \epsilon \) is set to 0.001 according to our repair method. Accordingly, if \( V_p \leq \epsilon \), corresponding constraint violations are determined to be acceptable and vice versa for cases 2–4. In addition, since the proposed repair method is easy to completely repair the constraint violations for the ELD problems without transmission losses, the smaller the constraint violations are, the finer corresponding solution is. If constraint violations are equal to 0, the feasible solution can be obtained.

In Table 6, the best results produced by PADE and the other methods from the literature for case 2 are presented. The best cost obtained by PADE is lower than those of QOTLBO [3], TLBO [3], and BSA [49], and it is higher than that of OGHS [20]. OGHS [20] obtains the lowest cost among all methods, but its \( \text{Cost} \) value is equal to 2.11, which is far larger than the error tolerance (\( \epsilon = 0.001 \)) for this case. Thus the solution obtained by OGHS [20] is infeasible. All \( \text{Cost} \) values obtained by PADE and the other methods are equal to 0, indicating that their inequality constraints are strictly met. Furthermore, the \( V_p \) values of PADE, QOTLBO [3], TLBO [3], and BSA [49] meet the error tolerance for this case, so their solutions are considered as the feasible ones. The best cost obtained by TLBO [3] is the largest on all the best costs obtained by the other methods producing the feasible solutions. In all, PADE outperforms the other methods in achieving the minimal fuel cost while meeting the restriction of constraints.

Regarding the optimization of case 3, Table 7 presents the comparison between PADE and the other methods reported in the literature. For cases 3-4, the \( \text{Cost}_{\text{mean}} \) value of PADE with phase-based mutation is slightly larger than that of PADE with DE/best/1 and is lower than that of PADE with other mutation operators. It means that the solutions obtained by PADE with phase-based mutation are slightly worse than those of PADE with DE/best/1 but are better than those of PADE with other mutation operators. PADE with phase-based mutation outperforms PADE with another five mutation operators on the solution quality and stability for the last two complex ELD problems.

### Table 4: Comparison between PADE with adaptive and fixed control parameters for six ELD problems.

| Problems | Terms | PADE with fixed control parameters | PADE with adaptive control parameters |
|----------|-------|-----------------------------------|--------------------------------------|
| Case 1   | \( \text{Cost}_{\text{mean}} \) | 15275.930392 | 15275.930392 |
|          | \( \text{Cost}_{\text{std}} \) | 5.393880e-12 | 5.393880e-12 |
| Case 2   | \( \text{Cost}_{\text{mean}} \) | 63975.710817 | 63975.710808 |
|          | \( \text{Cost}_{\text{std}} \) | 6.775580e-06 | 8.884254e-10 |
| Case 3   | \( \text{Cost}_{\text{mean}} \) | 111497.563739 | 111497.562806 |
|          | \( \text{Cost}_{\text{std}} \) | 4.724600e-04 | 9.790660e-05 |
| Case 4   | \( \text{Cost}_{\text{mean}} \) | 4303.508253  | 4303.508087  |
|          | \( \text{Cost}_{\text{std}} \) | 1.098271e-04 | 7.348985e-05 |
| Case 5   | \( \text{Cost}_{\text{mean}} \) | 121476.565088 | 121432.136531 |
|          | \( \text{Cost}_{\text{std}} \) | 22.458685     | 20.625706     |
| Case 6   | \( \text{Cost}_{\text{mean}} \) | 121485.558871 | 121433.128567 |
|          | \( \text{Cost}_{\text{std}} \) | 31.898932     | 20.061615     |
Table 5: The influence of phase-based mutation for six ELD problems.

| Problems | Operators | Cost<sub>mean</sub> | Cost<sub>std</sub> |
|----------|-----------|--------------------|------------------|
| Case 1   | PADE with DE/rand/1 | 15275.930392 | 5.570775e-12 |
| Case 1   | PADE with DE/rand/2 | 15275.930392 | 5.550256e-12 |
| Case 1   | PADE with DE/best/1 | 15275.930392 | 5.446504e-12 |
| Case 1   | PADE with DE/best/2 | 15275.930392 | 5.550256e-12 |
| Case 1   | PADE with DE/current-to-best/1 | 15275.930392 | 5.446504e-12 |
| Case 1   | PADE with phase-based mutation | 15275.930392 | 5.393880e-12 |
| Case 1   | PADE with DE/rand/1 | 63975.710809 | 1.398844e-07 |
| Case 1   | PADE with DE/rand/2 | 63975.710813 | 3.220744e-06 |
| Case 1   | PADE with DE/best/1 | 63975.710808 | 3.128049e-11 |
| Case 1   | PADE with DE/best/2 | 63975.710808 | 5.391686e-09 |
| Case 2   | PADE with DE/rand/1 | 111497.563242 | 2.967826e-04 |
| Case 2   | PADE with DE/rand/2 | 111497.563663 | 4.310065e-04 |
| Case 2   | PADE with DE/best/1 | 111497.563242 | 2.967826e-04 |
| Case 2   | PADE with DE/best/2 | 111497.563663 | 4.310065e-04 |
| Case 2   | PADE with DE/current-to-best/1 | 111497.563242 | 2.967826e-04 |
| Case 2   | PADE with phase-based mutation | 111497.563663 | 4.310065e-04 |
| Case 3   | PADE with DE/rand/1 | 4303.508224 | 6.472343e-05 |
| Case 3   | PADE with DE/rand/2 | 4303.508471 | 8.734597e-05 |
| Case 3   | PADE with DE/best/1 | 4303.508002 | 6.472343e-05 |
| Case 3   | PADE with DE/best/2 | 4303.508103 | 8.310080e-05 |
| Case 3   | PADE with DE/current-to-best/1 | 4303.508264 | 1.028932e-04 |
| Case 3   | PADE with phase-based mutation | 4303.508087 | 7.348985e-05 |
| Case 4   | PADE with DE/rand/1 | 121470.028620 | 27.830628 |
| Case 4   | PADE with DE/rand/2 | 121477.405869 | 26.157857 |
| Case 4   | PADE with DE/best/1 | 121470.028620 | 27.830628 |
| Case 4   | PADE with DE/best/2 | 121477.405869 | 26.157857 |
| Case 4   | PADE with DE/current-to-best/1 | 121470.028620 | 27.830628 |
| Case 4   | PADE with phase-based mutation | 121477.405869 | 26.157857 |
| Case 5   | PADE with DE/rand/1 | 121602.556257 | 131.422210 |
| Case 5   | PADE with DE/rand/2 | 121559.580775 | 110.238750 |
| Case 5   | PADE with DE/best/1 | 121456.571844 | 46.328972 |
| Case 5   | PADE with DE/best/2 | 121432.136531 | 20.061615 |
| Case 5   | PADE with DE/current-to-best/1 | 121432.136531 | 20.061615 |
| Case 5   | PADE with phase-based mutation | 121432.136531 | 20.061615 |
| Case 6   | PADE with DE/rand/1 | 121522.816073 | 79.358607 |
| Case 6   | PADE with DE/rand/2 | 121434.323771 | 22.678410 |
| Case 6   | PADE with DE/best/1 | 121433.128567 | 20.061615 |

Table 6: Comparison among different approaches for case 2.

| Unit j | QOTLBO [3] | TLBO [3] | OGHS [20] | BSA [49] | PADE |
|--------|------------|----------|-----------|----------|------|
| 1      | 79.5547    | 80.617   | 83.7483   | 79.6762  | 80.754354262 |
| 2      | 88.8977    | 92.4059  | 92.0583   | 88.7507  | 87.6905029999 |
| 3      | 210        | 210      | 210       | 210      | 210  |
| 4      | 224.9944   | 225      | 225       | 225      | 225  |
| 5      | 324.9708   | 324.9862 | 315       | 325      | 325  |
| 6      | 324.9977   | 320.1625 | 325       | 324.9927 | 325  |
| C      | 63977      | 64032    | 63953.08  | 63976    | 63975.710808 |
| V<sub>G</sub> | 0         | 0        | 0         | 0        | 0    |
| V<sub>P</sub> | 0.00012   | 2.25e-05 | 2.11      | 6.68e-05 | 9.999999957e-04 |
in the literature. For OGHS [20], its best cost is reported to be 111490. However, its actual cost is equal to 111497.612522 after careful verification, which is slightly higher than that of PADE. The best cost obtained by PADE is lower than those of the other methods, and it is equal to 111497.562722. Besides, the constraint violations of all these methods reach our error tolerance set, suggesting that all the solutions are feasible for case 3. Therefore, PADE can achieve the most satisfactory generation dispatch of case 3.

Table 8 lists the best results obtained by the three methods for case 4. Among the three methods, the cost obtained by PADE is the lowest, and it is equal to 4303.507984. For the constraint violations of the three methods, all of their $V_G$ values are equal to 0, and all of their $V_p$ values are smaller than 0.001, revealing that all three methods have strong ability in finding the feasible solution. Overall, the performance of PADE is the best compared to the other two methods.

The best results yielded by PADE and the other methods from the literature are shown in Table 9 for case 5. Most methods from the literature report the corresponding best solution, but unfortunately the best solution, $V_G$ and $V_p$ values of IFEP [2], is not available. We get their best cost from the IFEP [2], and it is equal to 122624.35, which is larger than that of PADE. The best cost obtained by $\theta$-MBA [50] is equal to 121412.5355, which seems similar to that of PADE. However, the value of $\theta$-MBA [50] is incorrect after verification based on the corresponding solution, and the precise cost is equal to 121578.483653, which is much larger than that of PADE. Except for the $\theta$-MBA [50] amended, the best cost obtained by FPSO [16] is also amended to 123860.094144. The best costs obtained by GA-DE-PS [17] and NAPSO [4] are slightly higher than that of PADE, and the best costs obtained by the remaining reported methods are far higher than that of PADE. The $V_G$ value of FPSO [16] is nonzero and equal to 461.85; the $V_G$ value of any of the other methods is exactly equal to 0. If the accuracy required is equal to 1e-05, the solutions obtained by $\theta$-MBA [50], THS [11], and PADE are acceptable. However, if the accuracy required is equal to 1e-06, the solutions obtained by $\theta$-MBA [50] and THS [11] are infeasible, and the solutions obtained by PADE are still feasible. Therefore, it depends on the accuracy required whether the solutions obtained by other methods are feasible. The $V_p$ values of FPSO [16], NAPSO [4], GA-DE-PS [17], THS [11], and $\theta$-MBA [50] are larger than that of PADE, indicating that PADE is better than these methods in achieving the feasible solution.

The proposed algorithm and several approaches from the literature are used to optimize case 6, and their best results are given in Table 10. The best costs obtained by NAPSO [4] and KHA-IV [12] are slightly higher than that of PADE, and the best costs obtained by the remaining reported methods are far higher than that of PADE, which suggests that PADE has stronger search ability compared to the other methods. PSO [4], FAPSO [4], and KHA-IV [12] do not report their best solutions and constraint violations. However, they are inferior to PADE in the best cost. On the obtained $V_p$ value, the $V_p$ value of PADE equal to 1.82e-12 is the lowest, and the $V_p$ value of $\theta$-MBA [50] equal to 72 is the largest, which cannot be neglected. All methods meet the power output limits and POZs completely. Based on the best cost and constraint violation, PADE is more efficient for case 6 compared to the other methods.

4.8. Comparison among Different Constraint Handling Methods. Finally, the selected six ELD cases are employed to test the efficiency of the proposed repair methods and the other reported repair methods on handling constraints, which mainly refers to the equality constraint handling methods. Extensive experiment is carried out to compare the performance of PADE with different constraint handling strategies. Due to difference of the selected problems, we separately propose two types of repair methods for the ELD problems without transmission losses and the ELD problems with transmission losses, and the two repair methods are
Table 8: Comparison among different approaches for case 4.

| Unit j | BSA [49] | MHSA [19] | PADE |
|--------|----------|-----------|------|
| 1      | 104.1756 | 52.066    | 104.115664 |
| 2      | 92.10999 | 123.913   | 92.211244 |
| 3      | 50       | 51.0307   | 50   |
| 4      | 50       | 64.1151   | 50.000002 |
| 5      | 50.0001  | 55.4468   | 50.000012 |
| 6      | 50       | 52.4076   | 50.000001 |
| 7      | 50       | 53.8972   | 50   |
| 8      | 50       | 50        | 50.000002 |
| 9      | 62.8778  | 56.992    | 62.896512 |
| 10     | 63.0931  | 75.5895   | 62.994068 |
| 11     | 62.6157  | 106.821   | 62.675866 |
| 12     | 177.6497 | 98.1268   | 177.626134 |
| 13     | 50       | 69.1359   | 50   |
| 14     | 50       | 50        | 50   |
| C      | 4303.5111| 4304.95   | 4303.507984 |
| $V_G$  | 0        | 0         | 0    |
| $V_p$  | 0.00012  | 0.00016   | 9.989949e-04 |

called PRM1 and PRM2, respectively. In addition, the penalty method (PM) is combined with PRM1 to further handle constraints for the ELD problems without transmission losses, which is represented as PM+PRM1. For the ELD problems with transmission losses, the obtained solution is feasible when $V_p \leq \varepsilon$, so PRM2 does not need to combine PM. The penalty method (PM), the penalty method based on the first repair method (PM+RM1) [10], and the penalty method based on the second repair method (PM+RM2) [52] are used to compare with PM+PRM1 on solving the ELD problems without transmission losses. It should be explained that PM does not need to combine with RM2 for case 1 and case 5, and PM is necessary to combine with RM2 to handle POZs for case 6. The penalty method (PM), the penalty method based on the third repair method (PM+RM3) [53], and the fourth repair method (RM4) [9] are used to compare with PRM2 on solving the ELD problems with transmission losses. RM4 also uses an approximate method to repair infeasible solutions, where its error tolerance is the same as that of PRM2, so PM is not combined with RM4. All constraint handling methods are separately incorporated into PADE, and the population size and maximum generation number are the same as the above experiments. The results of 30 independent experiments are separately listed in Tables 11-12, where the terms AOFV, OFVSTD, and TACV represent the average objective function value, standard deviation of the objective function values, and total average constraint violation, respectively.

It can be clearly seen that PADE based on PM+PRM1 performs better than PADE based on the other constraint-handling methods in AOFV and OFVSTD for the ELD cases without transmission losses, and PADE based on PRM2 also performs better than PADE based on the other constraint-handling methods in AOFV and OFVSTD for the ELD cases with transmission losses. It means that the proposed two repair methods make PADE stronger search ability and stable compared to the other constraint-handling methods. In contrast, the AOFVs and OFVSTDs of PM are the largest on all cases, indicating that the search ability and stable of PADE based on PM are poor. The TACVs of PM+PRM1 are lower than that of PM for case 1, case 5, and case 6, and the TACVs of PM+PRM1 are equal to 0 for case 1. Although the TACVs of PM+PRM1 are larger than those of PM+RM1 and (PM)+RM2 for cases 5-6, its TACVs will also reach desirable accuracy if the error tolerance is set to 0.00001 for cases 5-6. For cases 2-4, the TACVs of all these methods meet the requirement of accuracy, because the error tolerance is set to 0.001 for the three cases. Therefore, to some extent, the proposed two repair methods exhibit good efficiency in finding feasible solutions for the ELD problems. In all, PM+PRM1 and PRM2 are very competitive compared to the other constraint-handling methods.

5. Conclusions

We improve the original differential evolution (DE) algorithm to solve the ELD problems. First, an archive of storing successful individuals is set to have a greater chance to obtain the updated individuals. Second, a phase-based mutation operation is executed, where DE/srand/1 and DE/best/1 are adopted in the early and late phase, respectively. To determine the threshold at the end of the early phase, a success ratio and auxiliary generation index threshold are used together to divide the entire evolution phase into the early and late phase. The mutation operation is beneficial to balance the global and local search when the algorithm solves the ELD problems. Third, the two control parameters are adaptively generated for every individual. Different from other self-adaptive control parameters, the control parameters owned by the individuals have the ability to learn from the best
Table 9: Comparison among different approaches for case 5.

| Unit | IFEP [2] | FPSO [16] | GA-DE-PS [17] | THS [11] | NAPSO [4] | θ-MBA [50] | PADE |
|------|----------|-----------|---------------|----------|-----------|------------|------|
| 1    |         | 110.7998  | 110.8104      | 110.8018 | 112.246   | 110.7998   |      |
| 2    |         | 110.7998  | 111.3803      | 110.8    | 112.3141  | 110.7998   |      |
| 3    |         | 97.3999   | 97.4036       | 97.3999  | 97.8202   | 97.3999    |      |
| 4    |         | 179.73    | 179.7344      | 179.7331 | 179.7331  | 179.7331   |      |
| 5    |         | 169.38    | 87.8331       | 87.7997  | 91.7458   | 87.7999    |      |
| 6    |         | 142.77    | 139.9999      | 140      | 140       | 140        |      |
| 7    |         | 334.37    | 259.5998      | 259.6021 | 259.6055  | 259.5965   |      |
| 8    |         | 284.6     | 284.6089      | 284.5996 | 284.6495  | 284.5965   |      |
| 9    |         | 284.6     | 284.4672      | 284.5996 | 284.6061  | 284.5965   |      |
| 10   |         | 273.39    | 130           | 130      | 130       | 130        |      |
| 11   |         | 240.87    | 94.0011       | 168.799  | 94        | 243.5996   |      |
| 12   |         | 310.28    | 94            | 94.0014  | 94        | 168.7997   |      |
| 13   |         | 394.18    | 214.7608      | 214.7597 | 125       | 214.7597   |      |
| 14   |         | 393.2     | 394.2831      | 394.2973 | 304.5195  | 394.2937   |      |
| 15   |         | 214.76    | 304.5142      | 394.2973 | 394.2976  | 394.2937   |      |
| 16   |         | 394.28    | 394.2784      | 394.2973 | 304.5196  | 394.2937   |      |
| 17   |         | 489.28    | 489.2793      | 489.2793 | 489.2798  | 489.2793   |      |
| 18   |         | 489.28    | 489.2793      | 489.2793 | 489.2794  | 489.2793   |      |
| 19   |         | 511.27    | 511.2793      | 511.2793 | 511.2794  | 511.2793   |      |
| 20   |         | 421.52    | 511.2795      | 511.2802 | 511.2793  | 511.2792   |      |
| 21   |         | 343.77    | 523.2797      | 523.2813 | 523.2793  | 523.2798   |      |
| 22   |         | 433.5     | 523.2797      | 523.2806 | 523.2793  | 523.2793   |      |
| 23   |         | 433.52    | 523.2793      | 523.2822 | 523.2793  | 523.2804   |      |
| 24   |         | 523.28    | 523.2796      | 523.2793 | 523.2796  | 523.2793   |      |
| 25   |         | 343.76    | 523.2796      | 523.2831 | 523.2793  | 523.2793   |      |
| 26   |         | 433.52    | 523.2796      | 523.2801 | 523.2793  | 523.2793   |      |
| 27   |         | 10        | 10.0002       | 10       | 10        | 10         |      |
| 28   |         | 9.55      | 10.0003       | 10       | 10        | 10         |      |
| 29   |         | 10        | 10            | 10       | 10        | 10         |      |
| 30   |         | 128.59    | 87.8003       | 97       | 87.7998   | 87.7990    |      |
| 31   |         | 259.47    | 189.9999      | 190      | 190       | 190        |      |
| 32   |         | 159.74    | 189.9999      | 190      | 190       | 190        |      |
| 33   |         | 259.47    | 189.9999      | 190      | 190       | 190        |      |
| 34   |         | 164.79    | 164.8002      | 164.8039 | 164.8014  | 164.7998   |      |
| 35   |         | 164.8    | 164.4059      | 194.3927 | 192.1066  | 194.3978   |      |
| 36   |         | 239.58    | 199.9997      | 199.4622 | 200       | 200        |      |
| 37   |         | 25        | 109.9998      | 110      | 109.9999  | 110        |      |
| 38   |         | 7.05      | 109.9999      | 110      | 109.9999  | 110        |      |
| 39   |         | 57.05     | 109.9999      | 110      | 109.9999  | 110        |      |
| 40   |         | 511.28    | 511.278       | 511.2784 | 511.2793  | 511.2794   |      |
| C    | 122624.35  | 123860.22  | 121412.8002  | 121425.15 | 121412.57 | 121412.555 |      |
| AC   | –         | 123860.09444 | –              | –       | –          | 121578.483653 | – |
| VG   | NA        | 461.85    | 0             | 0        | 0          | 0          | 0    |
| Vs   | NA        | 0.02      | 0.0011        | 0.0002   | 0.0015     | 0.0001     | 4e-06 |

*“-” represents that the presented C does not need to be amended.*

control parameters owned by the global best individual, which further enhances PADE's robustness. In addition to the algorithm's improvement, two types of repair methods are used to meet the requirement of the equality constraint for the ELD problems without or with transmission losses, respectively. As a result, the proposed repair methods can completely eliminate the constraint violation on solving some ELD cases or reduce a lower level compared to the ones for the other ELD cases. Due to the efficient repair methods, PADE can obtain a lot of high-quality feasible solutions. Finally,
| Unit \( j \) | NAPSO [4] | PSO [4] | FAPSO [4] | MHSA [19] | KHA-IV [12] | \( \theta \)-MBA [50] | PADE |
|---------|---------|---------|---------|---------|---------|---------|---------|
| 1       | 110.7997 | NA      | NA      | 112.2539 | NA      | 112.7998 | 110.799829 |
| 2       | 110.8004 | NA      | NA      | 113.8236 | NA      | 110.7998 | 110.799825 |
| 3       | 97.3971  | NA      | 97.8091 | 97.39991 | NA      | 97.39991 | 97.399913  |
| 4       | 179.7331 | NA      | 179.1497| NA      | 179.7331| 179.7331 | 179.7331   |
| 5       | 87.8009  | NA      | 95.9375 | NA      | 87.7999 | 87.7999  | 87.799905  |
| 6       | 139.9999 | NA      | 140     | NA      | 140     | 140     | 140       |
| 7       | 259.5996 | NA      | 260.6374| NA      | 259.5997| 259.59965| 259.59965  |
| 8       | 284.5961 | NA      | 284.7028| NA      | 284.5997| 284.59965| 284.59965  |
| 9       | 284.5997 | NA      | 284.5306| NA      | 284.5997| 284.59965| 284.59965  |
| 10      | 130      | NA      | 198.3517| NA      | 130     | 130     | 130       |
| 11      | 94       | NA      | 168.7328| NA      | 168.7998| 94      | 94        |
| 12      | 94       | NA      | 169.5461| NA      | 94      | 94      | 94        |
| 13      | 214.7597 | NA      | 125     | NA      | 125     | 214.7597| 214.7597  |
| 14      | 394.27937| NA      | 394.1953| NA      | 400     | 394.27937| 394.27937 |
| 15      | 394.27937| NA      | 305.965 | NA      | 394.2794| 394.27937| 394.27937 |
| 16      | 394.27937| NA      | 395.1542| NA      | 394.2794| 394.27937| 394.27937 |
| 17      | 489.27937| NA      | 491.4043| NA      | 489.2794| 489.27937| 489.27937 |
| 18      | 489.27937| NA      | 488.8733| NA      | 489.2794| 489.27937| 489.27937 |
| 19      | 511.27937| NA      | 511.3804| NA      | 511.2794| 511.27937| 511.27937 |
| 20      | 511.27937| NA      | 511.2114| NA      | 511.2794| 511.27937| 511.27937 |
| 21      | 523.2793 | NA      | 523.823 | NA      | 511.2794| 523.27937| 523.27937 |
| 22      | 523.2793 | NA      | 524.1971| NA      | 511.2794| 523.27937| 523.27937 |
| 23      | 523.2793 | NA      | 524.4822| NA      | 511.2794| 523.27937| 523.27937 |
| 24      | 523.2793 | NA      | 523.8066| NA      | 511.2794| 523.27937| 523.27937 |
| 25      | 523.2793 | NA      | 522.9924| NA      | 511.2794| 523.27937| 523.27937 |
| 26      | 523.2793 | NA      | 523.2932| NA      | 511.2794| 523.27937| 523.27937 |
| 27      | 10       | NA      | 10.3634 | NA      | 10      | 10      | 10        |
| 28      | 10       | NA      | 10.4596 | NA      | 10      | 10      | 10        |
| 29      | 10       | NA      | 11.445 | NA      | 10      | 10      | 10        |
| 30      | 87.7998  | NA      | 88.9212 | NA      | 91.43702| 87.799905| 87.799905 |
| 31      | 189.9999 | NA      | 190     | NA      | 190     | 190     | 190       |
| 32      | 189.9999 | NA      | 190     | NA      | 190     | 190     | 190       |
| 33      | 189.9999 | NA      | 189.8936| NA      | 190     | 190     | 190       |
| 34      | 164.8003 | NA      | 166.5926| NA      | 164.7998| 164.799825| 164.799825 |
| 35      | 194.4019 | NA      | 164.8243| NA      | 200     | 194.397778| 194.397778 |
| 36      | 199.9999 | NA      | 164.8891| NA      | 200     | 200     | 200       |
| 37      | 109.9999 | NA      | 109.5147| NA      | 110     | 110     | 110       |
| 38      | 109.9999 | NA      | 110     | NA      | 110     | 110     | 110       |
| 39      | 109.9999 | NA      | 109.7737| NA      | 110     | 110     | 110       |
| 40      | 511.2793 | NA      | 512.0693| NA      | 511.2794| 511.27937| 511.27937 |

\( C \)  | 121412.6102 | 123861.45 | 121719.73 | 121690.271 | 121412.5991 | 121491.0662 | 121412.535602 |

\( AC \) | \(-\)      | \(-\)    | \(-\)    | \(-\)    | \(-\)    | \(-\)    | 121522.397763 |

\( V_G \) | 0          | 0        | 0        | 0        | 0        | 0        | 0          |

\( V_{POZ} \) | 0          | 0        | 0        | 0        | 0        | 0        | 0          |

\( V_p \)  | 0.0018     | NA       | 0.0001   | NA       | 72       | 1.82e-12  |
Table 11: Comparison among different constraint handling methods for the ELD problems without transmission losses.

| Constraint handling methods | Terms   | Case 1          | Case 5          | Case 6          |
|-----------------------------|---------|-----------------|-----------------|-----------------|
| PM                          | AOFV    | 15322.203667    | 147671.17036    | 146645.192109   |
|                             | OFVSTD  | 27.29           | 6803.57         | 710.872         |
|                             | TACV    | 1.2e-11         | 0.0015          | 0.0011          |
|                             | AOFV    | 15275.930392    | 121535.444592   | 121622.856479   |
| PM+RM1                      | OFVSTD  | 5.45e-12        | 95.59           | 241.93          |
|                             | TACV    | 0               | 0               | 0               |
|                             | AOFV    | 15275.930392    | 121503.42274    | 123269.617648   |
| (PM)+RM2                    | OFVSTD  | 7.07e-12        | 61.4            | 566.56          |
|                             | TACV    | 0               | 0               | 0               |
|                             | AOFV    | 15275.930392    | 121432.136531   | 121433.128567   |
| PM+PRM1                     | OFVSTD  | 5.39e-12        | 20.63           | 20.06           |
|                             | TACV    | 0               | 4e-06           | 1.82e-12        |

Table 12: Comparison among different constraint handling methods for the ELD problems with transmission losses.

| Constraint handling methods | Terms   | Case 2          | Case 3          | Case 4          |
|-----------------------------|---------|-----------------|-----------------|-----------------|
| PM                          | AOFV    | 65197.353919    | 115279.671017   | 4570.409455     |
|                             | OFVSTD  | 494.82          | 902.82          | 77.76           |
|                             | TACV    | 0               | 0               | 0               |
|                             | AOFV    | 64101.996686    | 111851.899442   | 4410.372638     |
| PM+RM3                      | OFVSTD  | 146.86          | 174.21          | 17.51           |
|                             | TACV    | 0               | 0               | 0               |
|                             | AOFV    | 63975.710809    | 111497.563378   | 4303.508257     |
| RM4                         | OFVSTD  | 5.59e-07        | 0.00031         | 0.00011         |
|                             | TACV    | 0               | 0               | 0               |
|                             | AOFV    | 63975.710808    | 111497.562806   | 4303.508087     |
| PRM2                        | OFVSTD  | 8.88e-10        | 9.79e-05        | 7.35e-05        |
|                             | TACV    | 9.99999957e-04  | 9.999978e-04    | 9.989949e-04    |

PADE and the other five DE variants are used to optimize six ELD problems. The experimental results show that PADE performs better than, or as well as the other five state-of-the-art DE approaches in all cases. PADE is useful for finding better solutions in most cases while maintaining a low level for the constraint violations compared to the methods from the literature.

Data Availability

The data used to support the findings of this study will be provided by the author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [Grant no. 61403174] and the Postgraduate Research & Practice Innovation Program of Jiangsu Province [Grant no. KYCX17_1575].

References

[1] J. C. Dodu, P. Martin, A. Merlin, and J. Pouget, “Optimal formulation and solution of short-range operating problems for a power system with flow constraints,” Proceedings of the IEEE, vol. 60, no. 1, pp. 54–63, 1972.

[2] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, “Evolutionary programming techniques for economic load dispatch,” IEEE Transactions on Evolutionary Computation, vol. 7, no. 1, pp. 83–94, 2003.

[3] P. K. Roy and S. Bhui, “Multi-objective quasi-oppositional teaching learning based optimization for economic emission load dispatch problem,” International Journal of Electrical Power & Energy Systems, vol. 53, pp. 957–948, 2013.

[4] T. Niknam, H. D. Mojarrad, and H. Z. Meymand, “Non-smooth economic dispatch computation by fuzzy and self adaptive particle swarm optimization,” Applied Soft Computing, vol. 11, no. 2, pp. 2805–2817, 2011.

[5] J. P. D. Chattopadhyay, “A multi-area linear programming approach for analysis of economic operation of the Indian power system,” IEEE Transactions on Power Systems, vol. 11, no. 1, pp. 52–58, 1996.

[6] J. Y. Fan and L. Zhang, “Real-time economic dispatch with line flow and emission constraints using quadratic programming,”
M. Singh and J. S. Dhillon, “Multiobjective thermal power dispatch based on optimal lambda using radial basis function network,” International Journal of Electrical Power & Energy Systems, vol. 24, no. 7, pp. 551–556, 2002.

Z.-X. Liang and J. D. Glover, “A zoom feature for a dynamic programming solution to economic dispatch including transmission losses,” IEEE Transactions on Power Systems, vol. 7, no. 2, pp. 544–550, 1992.

Q. Qin, S. Cheng, X. Chu, X. Lei, and Y. Shi, “Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization,” Applied Soft Computing, vol. 59, pp. 229–242, 2017.

L. D. S. Coelho and V. C. Mariani, “An improved harmony search algorithm for power economic load dispatch,” Energy Conversion and Management, vol. 50, no. 10, pp. 2522–2526, 2009.

M. A. Al-Bitar, M. A. Awadallah, A. T. Khader, and A. L. Bolaji, “Tournament-based harmony search algorithm for non-convex economic load dispatch problem,” Applied Soft Computing, vol. 47, pp. 449–459, 2016.

B. Mandal, P. K. Roy, and S. Mandal, “Economic load dispatch using krill herd algorithm,” International Journal of Electrical Power & Energy Systems, vol. 57, pp. 1–10, 2014.

D. Zou, S. Li, Z. Li, and X. Kong, “A new global particle swarm optimization for the economic emission dispatch with or without transmission losses,” Energy Conversion and Management, vol. 139, pp. 45–70, 2017.

M. Basu, “Economic environmental dispatch using multi-objective differential evolution,” Applied Soft Computing, vol. 11, no. 2, pp. 2845–2853, 2011.

T. T. Nguyen and D. N. Vo, “The application of one rank cuckoo search algorithm for solving economic load dispatch problems,” Applied Soft Computing, vol. 37, pp. 763–773, 2015.

S. Kumar and D. K. Chaturvedi, “A fuzzy particle swarm optimization for solving the economic dispatch problem,” Advances in Intelligent and Soft Computing, vol. 130, no. 1, pp. 99–110, 2012.

B. Mahdad and K. Srairi, “Solving practical economic dispatch using hybrid GA-DE-PS method,” International Journal of Systems Assurance Engineering and Management, vol. 5, no. 3, pp. 391–398, 2014.

J. S. Alsumait, J. K. Sykulski, and A. K. Al-Othman, “A hybrid GA-PS-SQP method to solve power system valve-point economic dispatch problems,” Applied Energy, vol. 87, no. 5, pp. 1773–1781, 2010.

B. Jedd and V. Vahidinasab, “A modified harmony search method for environmental/economic load dispatch of real-world power systems,” Energy Conversion and Management, vol. 78, pp. 661–675, 2014.

M. Singh and J. S. Dhillon, “Multiobjective thermal power dispatch using opposition-based greedy heuristic search,” International Journal of Electrical Power & Energy Systems, vol. 82, pp. 339–353, 2016.

R. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Opposition-based differential evolution,” IEEE Transactions on Evolutionary Computation, vol. 12, no. 1, pp. 64–79, 2008.

R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

M. Basu, “Quasi-oppositional differential evolution for optimal reactive power dispatch,” International Journal of Electrical Power & Energy Systems, vol. 78, pp. 29–40, 2016.

D. Teijeiro, X. C. Pardo, D. R. Penas, P. Gonzalez, J. R. Banga, and R. Doallo, “A cloud-based enhanced differential evolution algorithm for parameter estimation problems in computational systems biology,” Cluster Computing, vol. 20, no. 3, pp. 1937–1950, 2017.

D. X. Zou, X. Wang, and N. Duan, “Iris location algorithm based on modified differential evolution algorithm,” Control Theory & Applications, vol. 30, no. 9, pp. 1194–1200, 2013.

N. M. Hamza, D. L. Essam, and R. A. Sarker, “Constraint Constrained Differential Evolution for Constrained Optimization,” IEEE Transactions on Evolutionary Computation, vol. 20, no. 3, pp. 447–459, 2016.

K. V. Price, R. M. Storn, and J. A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag, 2005.

W. Yi, Y. Zhou, L. Gao, X. Li, and J. Mou, “An improved adaptive differential evolution algorithm for continuous optimization,” Expert Systems with Applications, vol. 44, pp. 1–12, 2016.

S.-M. Guo, C.-C. Yang, P.-H. Hsu, and J.-S.-H. Tsai, “Improving differential evolution with a successful-parent-selecting framework,” IEEE Transactions on Evolutionary Computation, vol. 19, no. 5, pp. 717–730, 2015.

R. Tanabe and A. S. Fukunaga, “Improving the search performance of shade using linear population size reduction,” in Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’14), pp. 1658–1665, IEEE, July 2014.

A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” IEEE Transactions on Evolutionary Computation, vol. 13, no. 2, pp. 398–417, 2009.

J. Q. Zhang and A. C. Sanderson, “JADE: adaptive differential evolution with optional external archive,” IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 945–958, 2009.

M. Miranda-Varela and E. Mezura-Montes, “Constraint-handling techniques in surrogate-assisted evolutionary optimization. An empirical study,” Applied Soft Computing, vol. 73, pp. 215–229, 2018.

X. T. Li and M. H. Yin, “Modified differential evolution with self-adaptive parameters method,” Journal of Combinatorial Optimization, vol. 31, no. 2, pp. 546–576, 2016.

J. Liu, X. M. Yin, and X. S. Gu, “Differential evolution improved with adaptive control parameters and double mutation strategies,” in Proceedings of the Asian Simulation Conference/SCS Autumn Simulation Multi-Conference, pp. 186–198, 2016.

W. Yi, L. Gao, X. Li, and Y. Zhou, “A new differential evolution algorithm with a hybrid mutation operator and self-adapting control parameters for global optimization problems,” Applied Intelligence, vol. 42, no. 4, pp. 642–660, 2015.

L. X. Tang, Y. Dong, and J. Y. Liu, “Differential evolution with an individual-dependent mechanism,” IEEE Transaction on Evolution Computation, vol. 19, no. 4, pp. 560–574, 2015.

S. H. Wang, Y. Z. Li, H. Y. Yang et al., “Self-adaptive differential evolution algorithm with improved mutation strategy,” Soft Computing, vol. 22, no. 10, pp. 3433–3447, 2018.
[40] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Self-adaptive differential evolution incorporating a heuristic mixing of operators," *Computational optimization and applications*, vol. 54, no. 3, pp. 771–790, 2013.

[41] K. Thanushkodi, S. M. V. Pandian, R. S. D. Apragash et al., "An efficient particle swarm optimization for economic dispatch problems with non-smooth cost functions," *Wseas Transactions on Power Systems*, vol. 3, no. 4, pp. 257–266, 2008.

[42] M. Modiri-Delshad and N. A. Rahim, "Multi-objective backtracking search algorithm for economic emission dispatch problem," *Applied Soft Computing*, vol. 40, pp. 479–494, 2016.

[43] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[44] L. Jia, J. He, C. Zhang, and W. Gong, "Differential evolution with controlled search direction," *Journal of Central South University*, vol. 19, no. 12, pp. 3516–3523, 2012.

[45] D. T. Do, S. Lee, and J. Lee, "A modified differential evolution algorithm for tensile structures," *Composite Structures*, vol. 158, pp. 11–19, 2016.

[46] Y. Wang, Z. X. Cai, and Q. F. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.

[47] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," Technical Report, 2010.

[48] J. J. Liang, T. P. Runarsson, E. Mezura-Montes et al., "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," Technical Report, 2006.

[49] K. Bhattacharjee, A. Bhattacharya, and S. Halder Nee Dey, "Backtracking search optimization based economic environmental power dispatch problems," *International Journal of Electrical Power & Energy Systems*, vol. 72, pp. 830–842, 2015.

[50] A. Kavousi-Fard and A. Khosravi, "An intelligent $\theta$-modified bat algorithm to solve the non-convex economic dispatch problem considering practical constraints," *International Journal of Electrical Power & Energy Systems*, vol. 82, pp. 189–196, 2016.

[51] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[52] W. Sun, W. Shang, and D. Niu, "Application of improved ant colony optimization algorithm in distribution network planning," *Power System Technology*, vol. 30, no. 15, pp. 85–89, 2006.

[53] V. Hosseinnezhad and E. Babaei, "Economic load dispatch using $\theta$-PSO," *International Journal of Electrical Power & Energy Systems*, vol. 49, pp. 160–169, 2013.
