

## Research Article

# Adaptive Differential Evolution Based on Simulated Annealing for Large-Scale Dynamic Economic Dispatch with Valve-Point Effects

Dakuo He <sup>1,2</sup>, Le Yang <sup>1</sup>, and Zhengsong Wang<sup>1</sup>

<sup>1</sup>College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

<sup>2</sup>State Key Laboratory of Integrated Automation for Process Industries, Northeastern University, Shenyang, China

Correspondence should be addressed to Dakuo He; [hedakuo@mail.neu.edu.cn](mailto:hedakuo@mail.neu.edu.cn)

Received 12 May 2018; Revised 24 September 2018; Accepted 30 September 2018; Published 28 October 2018

Guest Editor: Eric Monfroy

Copyright © 2018 Dakuo He 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.

Dynamic economic dispatch (DED) that considers valve-point effects is a complex nonconvex and nonsmooth optimization problem in power systems. Over the past few decades, multiple approaches have been developed to solve this problem. In this paper, an adaptive differential evolution based on simulated annealing algorithm is proposed to solve the DED problem with valve-point effects. Simulated annealing (SA) algorithm is employed to carry out an adaptive selection mechanism in which the mutation operators of differential evolution (DE) are selected adaptively based on their historical performance. A mutation operator pool consisting of five operators is built to make each operator show its strength at different stages of the evolutionary process. Moreover, a heuristic strategy is introduced to transform infeasible solutions towards feasible ones to enhance the convergence rate of the proposed algorithm. The effectiveness of the proposed methods is demonstrated first on 10 popular benchmark functions with 100 dimensions, in comparison with the classic DE and five variants. Then, it is used to solve four DED problems with 10, 15, 30, and 54 units, which consider the valve-point effects, transmission loss, and prohibited operating zones. The simulation results are compared with those of state-of-the-art algorithms to clarify the significance of the proposed method and verify its performance. Three systems with 100-500 generators are also tested to confirm the advantages of the proposed method on large-scale DED problem.

## 1. Introduction

Dynamic economic dispatch (DED) is one of the most fundamental problems in modern power system operation and has been the topic of considerable research. The primary objective of DED is to minimize the total fuel cost and meet the load demand in a power system while satisfying system constraints. Solving a DED problem involves formulating a mathematical model of the problem and then selecting an appropriate optimization technique. Conventional mathematical programming based on several deterministic optimization techniques has been employed to solve such problems. These methods include the  $\lambda$ -iteration method, gradient method, and dynamic programming [1].

Solving a DED problem involves formulating a mathematical model of the problem and then selecting an

appropriate optimization technique. The methods developed to date can be categorized into conventional mathematical programming methods and heuristic-based optimization algorithms. Previous efforts to solve such problems have employed conventional mathematical programming based on several deterministic optimization techniques, such as the  $\lambda$ -iteration method [2], the gradient method [3], the Lagrange relaxation [4], and dynamic programming [5]. Considering the high nonlinearities and discontinuities in the real input-output characteristics of DED, these conventional methods often fail to adequately solve this problem. Considering the high nonlinearities and discontinuities in the real input-output characteristics of ED, these conventional methods often fail to solve this problem well.

Unlike conventional algorithms, modern heuristics optimization algorithms have the advantages of global search

capability and the lack of requirement for certain mathematical properties of the objective function to be satisfied. Thus, these algorithms have proven their effectiveness in solving complex and nonconvex problems such as DED [2]. Representative algorithms include the genetic algorithm (GA) [6], particle swarm optimization (PSO) [7], simulated annealing (SA) [8], artificial immune system (AIS) [9], artificial bee colony (ABC) [10], evolutionary programming (EP) [11], differential evolution (DE) [12], harmony search (HS) [13], seeker optimization algorithm (SOA) [14], imperialist competitive algorithm (ICA) [15], teaching-learning algorithm (TLA) [16], bacterial foraging optimization (BFO) [17], and others. In addition, to further enhance the global search ability of these original algorithms, representative modified variants, such as modified PSOs [18–21], modified DE [12, 22–24], modified ABCs [25–27], and others, have been developed over the past years. In addition to these innovations, hybrid methods that combine two or more approaches have been applied to solve this problem, such as EP-PSO [11], the hybrid genetic algorithm based on differential evolution (HGA-DE) [28], the hybrid PSO with sequential quadratic programming (PSO-SQP) [29], the hybrid GA-pattern search-SQP (GA-PS-SQP) [12], BCO-SQP [30], a modified EP-PSO (MHPEP-SQP) [19], and SOA-SQP [14]. By combining various approaches, the hybrid method overcomes a single component's limitations by applying other components' advantages to solve the DED problem. One combined approach has been applied to implement coarse searching, and another has been used for fine tuning in a hybrid method [31]. Although many of the aforementioned methods have provided high-quality solutions, they have their own limitations. Heuristic-based approaches are often trapped by the sensitivity of various parameter settings. With hybrid methods, finding an appropriate integration point for the combined approaches is always a challenging problem for designers. Some representative methods are listed in Table 1. The pros and cons of the studied works can be summarized by Table 2.

Differential evolution algorithm (DE) [32], which is proposed by Storn and Price, is a population based algorithm that simulates the biological evolution. Compared with other swarm intelligence algorithms, DE is a simple but powerful method for global numerical optimization with nonsmooth and nonconvex characteristics. Due to its simplicity and efficiency, DE has been successfully applied to many complex engineering fields, such as data mining [33], scheduling [34], energy distribution [35], molecular engineering [36], pattern classification [37], and so on.

However, DE still has some drawbacks. The successful application of DE mainly depends on its mutation operators and parameter settings. Moreover, DE often suffers the problem of premature convergence, which may lead the searching progress into local optimal. In addition, DE lacks a mechanism to deal with the complicated constraints of the DED problem [23].

In this paper, an adaptive differential evolution based on simulated annealing (ADE-SA) algorithm is proposed to solve the large-scale DED problem with valve-point effects. In ADE-SA, DE is employed to search the feasible solution of DED. Five widely used mutation operators are collected

to form a mutation operator pool. SA is employed to carry out an adaptive selection mechanism in which the mutation operators are selected adaptively according to their performance in the search history. Thus, individual can adaptively select appropriate mutation operators in the different stages of the search process. Furthermore, a heuristic strategy is introduced to transform infeasible solutions towards feasible ones to enhance the convergence rate of the proposed algorithm. The performance of ADE-SA is evaluated on a series of DED problems, and compared with other eight state-of-the-art algorithms. Comparison results indicate the effectiveness of the proposed algorithm. Three systems with 100-500 generators are also tested to confirm the advantages of the proposed method on large-scale DED problem.

The rest of this paper is organized as follows: Section 2 provides the formulation of the DED problem. The basic SA and DE scheme is briefly described in Section 3, while the framework of the proposed method is also presented in this Section. To illustrate its effectiveness, the proposed approach was tested on a series of systems in Section 4. Section 5 offers the conclusions. Parts of this paper, such as Sections 1 and 3.3.4, are the extended version of our completed research [38].

## 2. Problem Formulation

DED is a multiconstrained and nonlinear optimization problem. The primary objective of DED is to minimize the total fuel cost and meet the load demand in a power system while satisfying system constraints. Consider a power system with  $N_G$  generating units. In general, the DED problem can be formulated as the following optimal function [13]:

$$\min F = \sum_{t=1}^{N_T} \sum_{i=1}^{N_G} (a_i P_{i,t}^2 + b_i P_{i,t} + c_i) \quad (1)$$

where  $a_j$ ,  $b_j$ , and  $c_j$  are the fuel cost coefficients of generator  $j$ .  $N_T$  is the time horizon and  $N_G$  is the total number of generators of the system.  $P_{i,t}$  is the power output of generator  $i$  at time interval  $t$ .  $F$  denotes the total fuel costs over the dispatch period. When the valve-point effect (VPE) is considered, the total production costs can be written as follows [13]:

$$\begin{aligned} \min F \\ = \sum_{t=1}^{N_T} \sum_{i=1}^{N_G} (a_i P_{i,t}^2 + b_i P_{i,t} + c_i + |e_i \sin(f_i (P_{i,\min} - P_{i,t}))|) \end{aligned} \quad (2)$$

where  $e_j$  and  $f_j$  are the coefficients of generator  $i$  reflecting the valve-point effects.

The minimized DED problem should be subjected to the following equality and inequality constraints:

- (1) Real power balance constraints

$$\sum_{i=1}^{N_G} P_{i,t} - P_{D,t} = P_{L,t} \quad t \in N_T \quad (3)$$

$$P_{L,t} = \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} P_{i,t} B_{ij} P_{j,t} \quad t \in N_T \quad (4)$$

TABLE 1: Main features of representative methods in the area of DED problem.

Reference	Methods	Main features
[11]	EP-PSO	Evolutionary programming is applied as a based level search, and a local search sequential quadratic programming is used as a fine tuning to determine the optimal solution at the final.
[12]	IDE	Heuristic crossover technique and gene swap operator are introduced in the proposed approach to improve the convergence characteristic of the differential evolution algorithm.
[14]	SOA-SQP	SOA is used as a base level search, which can give a good direction to the optimal global region and SQP as a local search to fine tune the solution obtained from SOA.
[18]	IPSO	Feasibility-based rules and heuristic strategies with priority list based on probability are devised to handle constraints effectively.
[19]	DGPSO	PSO technique explores the solution space freely, and SQP will be called only when there is an improvement of solution (a feasible solution) in the PSO run.
[20]	TVAC-IPSO	Employing time varying acceleration coefficients iteration particle swarm optimization method.
[21]	ICPSO	By integrating chaotic optimization into particle swarm optimization, it can jump out of the local extreme zone quickly and improve particles' searching performance of the global best solution.
[22]	MDE	Feasibility-based selection comparison techniques and heuristic search rules are devised to handle constraints effectively.
[23]	CDE	Chaotic sequences are applied to obtain the dynamic parameter settings in DE, and a chaotic local search operation is designed to avoid premature convergence effectively.
[24]	CDE	It combines DE with the local search technique. DE is the main optimizer, while an approximated model for local search is applied to fine tune in the solution of the DE run.
[25]	MABC	It introduces a new relation to update the solutions within the search space to avoid premature convergence and to find stable and high quality solutions. Moreover, it is endowed with a chaotic sequence generated by both a cat map and a logistic map.
[26]	MBABC	It utilizes a measure of dissimilarity between binary strings for generating the new binary solutions for UCP. And an intelligent scout bee phase is proposed to replace the abandoned solution with the global best solution.
[27]	HSABC	
[28]	HGA-DE	GA is the main optimizer, while DE and SQP are used to fine tune in the solution of the GA run. The cost function of ED is approximated by using a smooth and differentiable function based on the maximum entropy principle.
[29]	PSO-SQP	It incorporates PSO as the main optimizer and SQP as the local optimizer to fine-tune the solution region whenever PSO discovers a better solution region in the progress of its run.
[30]	BCO-SQP	It incorporates bee colony optimization as a base level search which can give a good direction to the optimal region and sequential quadratic programming as a local search procedure which is used to fine tune that region for achieving the final solution.

TABLE 2: The pros and cons of the studied works.

Classification	Pros	Cons
Conventional methods	(a) Computationally efficient. (b) guarantee the optimization convergence and classically do not have any problem-specific parameters to specify (c) only need to run once	(a) Deal mainly with convex cost functions. (b) non-linear and non-smooth characteristics (c) Fail to generate high quality solutions.
Heuristic-based methods	(a) Do not require certain mathematical properties of the objective function. (b) Be good at dealing with non-convex and non-linear problems.	(a) sensitive to parameter settings (b) solution is not unique for each trial (c) prone to get struck into local minima. (d) lack the guarantee of convergence in finite time especially for large-scale DED problem
Hybrid methods	(a) Overcomes a single component's limitations. (b) Usually one combined method for coarse searching and the other for fine tuning.	An appropriate integration point of two algorithms is very difficult to determine.

where  $P_{D,t}$  and  $P_{L,t}$  are the total load demand and the system transmission loss of time interval  $t$ , which is expressed as a quadratic function using B coefficients [31].

(2) Power operating constraints

$$P_{i,\min} \leq P_{i,t} \leq P_{i,\max}, \quad i \in N_G, t \in N_T \quad (5)$$

where  $P_{i,\max}$  is the maximum power outputs of generator  $i$ .

(3) Generator ramp rate constraints

$$P_{i,t-1} - DR_i \leq P_{i,t} \leq P_{i,t-1} + UR_i, \quad i \in N_G \quad (6)$$

where  $UR_i$  and  $DR_i$  are the ramp-up and ramp-down rate constraints of generator  $i$ , respectively.

### 3. Methods

**3.1. Brief Introduction of Simulated Annealing Algorithm.** Simulated annealing (SA) [49] is a metaheuristic that can escape from local optimal with its use of mountain-climbing move. SA is a popular technique over the past 20 years due to its ease of implementation and convergence characteristics. It is usually used for solving discrete and continuous optimization problems. SA comes from the imitation of the physical annealing process. In this process, the crystalline solid is heated and then cooled slowly until it reaches its most stable molecular structure.

At each iteration of SA, the current solution and the new solution are compared. The improved solution is always accepted, while a part of the nonimproved (inferior) solution is also accepted to make the individual escape local optimum. The probability of accepting a nonimproved solution depends on a temperature parameter which is presented in

$$p = \begin{cases} 1 & \text{if } dif < 0 \\ e^{(-dif/T)} & \text{if } dif \geq 0 \end{cases} \quad (7)$$

where  $dif$  is the difference between the new solution and the original solution.  $T$  is the current temperature.

SA reaches steady state through a slow drop in temperature, that is, the temperature determines when the iteration terminates. Therefore, a temperature cooling function is needed to reduce the temperature from high temperature to high temperature. The cooling function is used to lower the temperature which is defined as

$$T = \frac{T}{1 + \beta T} \quad (8)$$

where  $\beta$  is a positive number less than 1.

**3.2. Brief Introduction of Differential Evolution.** The basic DE consist of three phases: mutation, crossover, and selection. In the mutation phase DE randomly selects three distinct individuals  $x_{r1}^G$ ,  $x_{r2}^G$ , and  $x_{r3}^G$  from the population such that  $i \neq r1 \neq r2 \neq r3$ .  $G$  represents the current generation. These three random individuals are employed to generate a mutated individual  $v_i^G$  according to

$$v_i^G = x_{r1}^G + F(x_{r2}^G - x_{r3}^G) \quad (9)$$

where  $F > 0$  is a scaling factor.

After the mutation phase, DE introduces crossover operation to increase the diversity of the population. The offspring individual  $u_i^G$  is generated from its parents  $x_i^G$  and the mutated individual  $v_i^G$  using the following crossover rule:

$$u_{i,j}^G = \begin{cases} v_{i,j}^G & \text{if } R_j \leq C_r \text{ or } j = rn(i) \\ x_{i,j}^G & \text{otherwise} \end{cases} \quad (10)$$

where  $C_r$  is a crossover constant  $\in [0, 1]$  and  $j$  denotes the  $j$ th element of individual.  $R_j \in [0, 1]$  is a uniformly distributed random number and  $rn(i) \in \{1, \dots, N\}$  is a randomly chosen index which guarantees that  $u_i^G$  gets at least one element from the mutated individual  $v_i^G$ .

In the selection phase the fitness value at the individual,  $f(u_i^G)$ , is compared with  $f(v_i^G)$ ; the winner will be the

survivor. The selection operation can be expressed as the following rule:

$$x_i^{G+1} = \begin{cases} u_i^G & \text{if } f(u_i^G) \leq f(x_i^G) \\ x_i^G & \text{otherwise} \end{cases} \quad (11)$$

### 3.3. Adaptive Differential Evolution Based on Simulated Annealing (ADE-SA)

**3.3.1. Mutation Operator Pool.** It is well-known that DE needs relatively less parameter tuning than many other swarm intelligence algorithms. In the mutation phase, the new individuals are generated by selecting three individuals randomly, which may cause over diversity in population and slow convergence. According to [50], DE can make use of its adaptive mutation strategy to improve its performance. Therefore, a mutation operator pool composed of five different mutation operators is built in this section. These operators will be implemented under the guidance of an adaptive selection mechanism at different stages of search. Through the analysis of the widely used evolutionary strategies in swarm intelligence algorithms, a total of 5 mutation operators are selected to make up the mutation pool (the first operator M1 is presented in (9)), which are as follows:

M2. DE/rand/2

$$v_i^G = x_{r1}^G + F(x_{r2}^G - x_{r3}^G) + F(x_{r4}^G - x_{r5}^G) \quad (12)$$

M3. DE/best/1

$$v_i^G = x_{\text{best}}^G + F(x_{r1}^G - x_{r2}^G) \quad (13)$$

M4. DE/current-to-best/1 derived from PSO

$$v_i^G = x_i^G + F(x_{\text{best}}^G - x_i^G) + F(x_{r1}^G - x_{r2}^G) \quad (14)$$

M5. Employed bees phase in ABC

$$v_i^G = x_i^G + \phi(x_{r1}^G - x_{r2}^G) \quad (15)$$

where  $r1, r2, r3, r4,$  and  $r5$  are mutually integers of randomly selected food sources except for  $i$ .  $F$  is the difference vector, and  $\phi$  is a random number in the range of  $[-1, 1]$  in ABC.

**3.3.2. Adaptive Selection Mechanism Based on Simulated Annealing.** In fact, each mutation operator may shows different performance levels at different stages of search process. Operators that show effectiveness in some stages of the iterative process may not perform well in other stages. In this section, a learning cycle is utilized to guide the adaptive selection of these operators. It records the historical performance of each mutation operators and calculates the chosen probability of each operator for the next cycle. The learning cycle should be much smaller than the whole iteration process to reflect the different stages of the search process. The learning cycle, which is proposed in [51], is employed in this paper. The details of the learning cycle are presented as follows.

*Step 1.* Update the total calls of all mutation operators  $c_i^{\text{total}}$ .

*Step 2.* Evaluate the new individual  $x'$  and the original individual  $x$ , and calculate the difference of their fitness functions,  $dif = f(x') - f(x)$ .

*Step 3.* Use the Metropolis acceptance criterion to decide whether to accept a new individual and update the parameter  $c_i^{\text{accept}}$  and  $C_a$  according to the result.

*Step 4.* Check if the reheating phase ( $C_a/LC < r_e$ ) is triggered, update the parameters  $t_{\text{imp}}$  and calculate the chosen probability of each operator. If not, only calculate the chosen probability.

*Step 5.* If the iterative process completes a learning cycle, terminate the loop. If not, return to Step 1.

Where  $c_i^{\text{total}}$  is the total number of calls of an operator during the current learning cycle,  $c_i^{\text{accept}}$  is the number of individuals that passed the Metropolis acceptance criterion.  $C_a$  counts the total number of accepted operator calls in a learning cycle.  $r_e$  is the stopping acceptance ratio which is used to trigger the reheating phase of SA. The temperature will be increased to a former temperature. This process continues until a better solution is found. The temperature is increase using

$$T = \frac{T}{1 - \beta T} \quad (16)$$

**3.3.3. Framework of ADE-SA.** The framework of ADE-SA can be summarized as follows.

*Step 1.* Set initial nonimproving acceptance ratio  $r_s$  and stopping nonimproving acceptance ratio  $r_e$ . Estimate the starting temperature  $T_s$ , stopping temperature  $T_e$ , total iterations  $iterMax$ , number of iterations for DENiter, and the length of single learning cycle  $LC$ . Set,  $t_{\text{imp}} = T_s$ ,  $iter = 0$ , and  $Ca = 0$ .

*Step 2.* Define the mutation strategy pool; initialize the associate parameters  $c_i^{\text{accept}} = 0$ ,  $c_i^{\text{total}} = 0$ , minimal weight  $w_{\text{min}}$ , and an initial weight  $w_i = w_{\text{min}}$ .

*Step 3.* Generate the initial population and calculate the fitness value based on the fitness function.

*Step 4.* Select an operator  $M_i$  based on  $p_i = w_i / \sum_{i=1}^n w_i$  and generate a candidate solution using operator  $M_i$ .

*Step 5.* Judge whether the candidate solution is acceptable or not based on the Metropolis criterion in SA.

*Step 6.* Check the reheating condition; if necessary, increase the temperature according to the function  $t = t/(1 - \beta t)$ .

*Step 7.* Check if  $niter$  is satisfied, cool the temperature based on  $t = t/(1 + \beta t)$ .

*Step 8.* Record the history information of current low-level heuristics and check if the learning period is meet, if so, update the associate parameters.

*Step 9.* If the stopping criteria satisfied, terminate the main loop. If not, return to Step 3.

**3.3.4. Heuristic Constraint Handling.** DED is a nonlinear constrained optimization problem with many equality and inequality constraints. Some solutions will not satisfy all constraints during the iterative process, such as the real power balance and ramp rate constraints. Penalty method is one of the most frequent-used approaches for complex constraints of DED, but the method is not suitable for improving the computational efficiency. In this section, a heuristic procedure has been employed to accommodate constraints [38]. Taking a solution  $P$  as an example, the following mechanism is applied to transform an infeasible solution into feasible solution.

*Step 1.* Randomly select an hour  $t \in N_T$  as the initial time interval.

*Step 2.* The power operating constraints are addressed based on the ramp rate constraints as follows:

$$P_{i,t}^{min} = \begin{cases} \max(P_{i,min}, P_{i,t+1} - UR_i), & t = 1 \\ \max(P_{i,min}, P_{i,t-1} - DR_i), & t = N_T \\ \max(\max(P_{i,min}, P_{i,t-1} - DR_i), P_{i,t+1} - UR_i), & t \in N_T \end{cases} \quad (17)$$

$$P_{i,t}^{max} = \begin{cases} \min(P_{i,max}, P_{i,t+1} + DR_i), & t = 1 \\ \min(P_{i,max}, P_{i,t-1} + UR_i), & t = N_T \\ \min(\min(P_{i,max}, P_{i,t+1} + DR_i), P_{i,t-1} + UR_i), & t \in N_T \end{cases} \quad (18)$$

Note that even if a feasible solution  $P_t$  is obtained at current hour, it may make it impossible for  $P_{t+1}$  to meet (3) and (6) at the next hour. Equations (17) and (18) can make sure that the generation range of the adjacent time interval satisfies the load demand. This prevention strategy helps to avoid generating a potential  $P_t$  which affects the equality constraints adjustment at the next hour.

*Step 3.* Adjust the output of each generator at current time interval to satisfy (5) and (6) using

$$P_{i,t} = \begin{cases} P_{i,t}^{min}, & P_{i,t} < P_{i,t}^{min} \\ P_{i,t}, & P_{i,t}^{min} \leq P_{i,t} \leq P_{i,t}^{max} \\ P_{i,t}^{max}, & P_{i,t} > P_{i,t}^{max} \end{cases} \quad (19)$$

*Step 4.* Tackle the demand-supply constraint by the following steps.

*Step 4.1.* Calculate the system transmission loss at the current time interval using (4).

*Step 4.2.* Calculate the difference in the power demand constraints at the current time interval as

$$Dif = P_{D,t} + P_{L,t} - \sum_{i=1}^{N_G} P_{i,t} \quad (20)$$

Check if  $Dif \leq \epsilon_0$ , go to **Step 2** and set  $t = t+1$ ; otherwise, go to the next step. Here,  $\epsilon_0$  is a tolerance variable which has a large value at the early stages of evolutionary process and is decreased gradually to a small value [31].

*Step 4.3.* Obtain a random generator  $r$  from  $N_G$  generators as the slack generator, and reset its output at current interval as

$$P_{r,t} = \begin{cases} P_{r,t} + Dif & \text{if } Dif > 0 \\ P_{r,t} - Dif & \text{if } Dif < 0 \end{cases} \quad (21)$$

*Step 4.4.* Adjust the output  $P_{r,t}$  to the feasible horizon using (18). If the solution of  $P_t$  is still infeasible, recalculate the  $Dif$  using (19) and select another generator  $r$  at random from  $N_G$  generators as the slack generator. Repeat this process until a feasible solution is found. Then update the operating range using (20) and (21), and set  $t = t + 1$ .

*Step 4.5.* Repeat **Step 4.1** to **Step 4.4** until  $t = N_T$ . Here we will obtain a feasible solution at  $t$  to  $N_T$  time intervals.

*Step 5.* Obtain the solution at the rest of the time intervals by repeating **Step 4.1** to **Step 4.4**. The strategy is similar to the preceding process, except the solution is calculated from the initial time interval  $t$  to the  $l$ th time interval.

*Step 6.* If  $t = 1$ , terminate the above process and output the updated feasible solution  $x$ .

## 4. Numerical Experiments and Results

To assess the efficiency of the proposed approach, 10 widely used benchmark functions are applied in this section. ADESA is compared with several recent popular modified DE algorithms, including IDE [12], CDE [24], HDE [39], and IMMSADE [52], respectively.

**4.1. Benchmark Functions.** The details of 10 commonly used benchmark functions are given in Table 3. Table 3 lists the 10 benchmark functions that are selected: F1~F5 are unimodal functions. F6 has one minimum and is discontinuous. F7 is a noisy quadratic function. F8~F10 are multimodal functions.  $D$  represents the dimension of the benchmark function,  $S$  represents the search ranges of the solutions, and  $f_{min}$  is the value of the global optimum. 30 independent runs with a commonly used 300000 max number of function evaluations (FES) are conducted. The simulation parameters chosen in this paper are as follows:  $r_s = 0.1$ ,  $r_e = 0.01$ ,  $w_{min} = 0.1$ ,  $LC = 25$ , population size  $NP = 50$ ,  $F = 0.44$ , and  $Cr = 0.9$ .

**4.2. Simulation Results and Discussions.** The mean and standard variances obtained by each algorithm on each test

TABLE 3: Benchmark functions.

Formulation	Type	$D$	$S$	$f_{\min}$
$f_1(x) = \sum_{i=1}^D x_i^2$	Unimodal	100	$[-100, 100]$	0
$f_2(x) = x_1^2 + 10^6 \sum_{i=1}^D x_i^2$		100	$[-100, 100]$	0
$f_3(x) = \sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $		100	$[-10, 10]$	0
$f_4(x) = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2$		100	$[-100, 100]$	0
$f_5(x) = \max\{ x_i , 1 \leq i \leq D\}$		100	$[-10, 10]$	0
$f_7(x) = \sum_{i=1}^D ( x_i + 0.5 )^2$	Discontinuous	100	$[-10, 10]$	0
$f_{22}(x) = \sum_{i=1}^D ix_i^4 + \text{random}[0, 1)$	Noisy	100	$[-600, 600]$	0
$f_{23}(x) = \sum_{i=1}^D  x_i \sin(x_i) + 0.1x_i $	Multimodal	100	$[-10, 10]$	0
$f_{24}(x) = \sum_{i=1}^D (x_i^2/4000) - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$		100	$[-600, 600]$	0
$f_{25}(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^D x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^D x_i^2}$		100	$[-5.12, 5.12]$	0

functions with 100 dimensions are summarized in Table 4. It can be seen in Table 4 that ADE-SA outperformed all other modified DE algorithms on seven functions, including  $f_1, f_2, f_3, f_5, f_6,$  and  $f_7$ . ADE-SA have the same best results as well as HDE and IMMSADE on functions  $f_9$  and  $f_{10}$ , while they are both much better than other algorithms on these two test functions. For  $f_4$  and  $f_8$ , HED and IMMSADE achieved the best solutions, respectively. But they are just slightly better than ADE-SA. So we can draw a basic conclusion from Table 4 that ADE-SA is good at solving unimodal and multimodal functions, and it has the advantage of solving high dimension problems.

Figure 1 shows the convergence curve of the compared 7 algorithms on 4 representative benchmark functions with different types. We can easily find that the search accuracy and the convergence rate of ADE-SA are better than the classic DE and all the modified DE variants on those of different types of test functions. The effectiveness of the proposed algorithm is fully demonstrated according to the convergence characteristics, and ADE-SA also shows a stable robustness. Although in some case ADE-SA converged slightly slower than other DE variants, it could also find high-quality result in the later stage of iteration. Overall, the proposed method presents a competitive search capability and rapid convergence on most of the benchmark functions.

## 5. Experimental Results on DED Problems and Discussions

In this section, to assess the efficiency of the proposed approach, we consider several test problems that involve up to 500 units in DED problems.

*Case 1.* 10-unit system with and without power loss [31].

*Case 2.* 15-unit system with prohibited operation zones and power loss [53].

*Case 3.* 30-unit system generated by combining three 10-unit systems of Case 1.

*Case 4.* 54-unit system without power loss [46].

*Case 5.* 100-unit system generated by combining ten 10-unit systems of Case 1 [43].

*Case 6.* 200-unit system generated by combining twenty 10-unit systems of Case 1.

*Case 7.* 500-unit system generated by combining fifty 10-unit systems of Case 1.

In all cases, the valve-point effect and the ramp rate constraints are considered. The complete dispatch horizon  $N_T$  is selected as one day with 24 hours.

*5.1. Experimental Settings.* The following values are used to evaluate the quality of the solutions: best/minimum total fuel cost  $F(B)$ , average total fuel cost  $F(A)$ , worst/maximum total fuel cost  $F(W)$ , and standard deviation SD. The simulation parameters chosen in this paper are as follows:  $r_s = 0.1$ ,  $r_e = 0.01$ ,  $w_{\min} = 0.1$ ,  $LC = 25$ , population size  $NP = 50$ ,  $F = 0.44$ , and  $Cr = 0.9$ . The hourly load demands for case 1 are shown in Table 5 and the unit data are adapted from [15] and listed in Table 6. For all the cases, the simulation contains 50 independent runs to ensure the robustness of the proposed method. Simulations were executed using the MATLAB 2016a computational environment on a Pentium-core 3.2 GHz personal computer with 8 GB of RAM.

### 5.2. Result and Comparison

*Case 1.* 10-generator problem with and without considering the transmission loss.

The first power test system consists of ten thermal generators without considering the transmission loss. The generating system data were taken from [31] and given in Table 6. The hourly system load demand for this case is listed in Table 5. The optimal generation dispatch values of the case without and with considering the transmission loss are shown in Tables 7 and 9, respectively. Tables 8 and 10 give the corresponding statistic comparison among different heuristic methods; 20 previously developed algorithms were used for comparison. It is obvious that ADE-SA can obtain solutions with much lower total generation costs than other reported

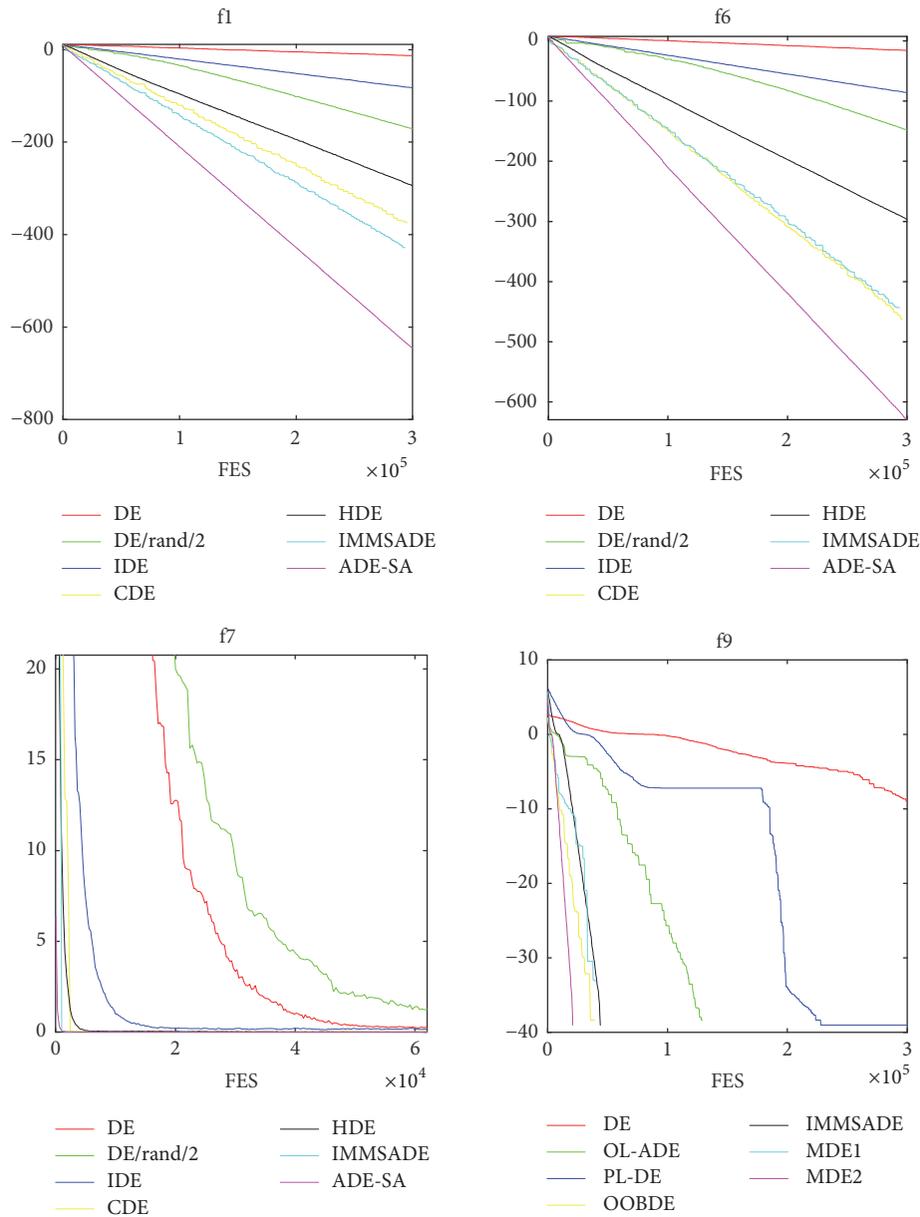


FIGURE 1: Comparison for average convergence curve of seven methods.

methods. For the case without transmission loss, although MIQP [31] can obtain lower standard deviation; ADE-SA outperforms it in terms of the total generation costs. Even the maximum costs obtained using these three methods are better than the solution obtained by MIQP.

*Case 2.* 15-generator problem with considering the prohibited operation zones and the transmission loss.

In this case, the power demand, operating unit characteristics, and B coefficients for calculation of transmission losses were given in [46]. The optimal generation dispatch values of the case with considering the transmission loss are shown in Table 11. From Table 12, it is obvious that ADE-SA gave the best performance on minimum and average fuel cost

compared with HS [46], HIS [46], GHS [46], SGHS [46], and NPAHS [46].

*Case 3.* 30-generator system generated by combining three 10-generator systems of Case 1.

The second case is a lossless electric power system consisting of 30 thermal generators which is tripled from the 10-generator problem of Case 1. The minimum, maximum, and average values of total operating cost for one day are given in Table 13, along with the operating costs from other published results from the literature for the sake of comparison. It is apparently seen that ADE-SA outperforms any other method regardless of minimum cost, average cost, and maximum cost.

TABLE 4: Unit data of Case 1.

Function	$f_1$		$f_2$		$f_3$		$f_4$		$f_5$	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
Method										
DE	3.93e-06	1.26e-12	5.06	4.25	0.0051	6.67e-06	1.27e+04	3.02e+07	1.48	0.14
DE/rand/2	4.59e-26	3.47e-31	1.24	0.98	0.0025	5.13e-07	1.01e+06	2.75e+09	4.12e-4	4.52e-09
IDE	5.44e-75	.96e-148	1.49e-68	2.05e-135	1.16e-31	1.35e-61	1.67e-43	2.80e-85	4.76e-47	2.27e-92
CDE	1.70e-36	3.42e-72	3000	2.33e+07	5.32e-20	1.58e-39	1.45e+03	2.52e+06	0.0013	5.69e-07
HDE	3.03e-163	0	5.66e-157	2.59e-312	2.63e-75	6.21e-149	2.24e-28	5.00e-55	2.18e-40	1.28e-79
IMMSADE	2.51e-128	6.34e-255	1.41e-129	1.98e-257	2.07e-65	2.83e-130	1.05e-63	1.11e-125	1.06e-70	9.32e-140
ADE-SA	1.81e-186	0	6.73e-176	0	2.83e-88	3.93e-175	1.20e-08	1.40e-15	7.85e-79	6.14e-116
	$f_6$		$f_7$		$f_8$		$f_9$		$f_{10}$	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
DE	2.02e-03	9.38e-07	1.058	1.58e-04	0.041	3.42 e-04	2.02e-04/	4.06e-07	1.89e+02	63.10
DE/rand/2	9.26e-31	1.97e-62	2.135	1.243	0.014	2.48 e-04	0	0	2.91	1.09
IDE	7.58e-05	2.00e-08	0.69	0.002	0.015	2.48 e-05	3.13e-35	7.84e-41	2.91	1.09
CDE	2.40e-32	1.53e-64	0.037	0.001	0.001	1.05 e-06	0	0	127.57	59.42
HDE	1.89e-12	/3.02e-24	0.017	0.001	1.76e-06	5.85e-12	0	0	0	0
IMMSADE	6.18e-04	5.52e-09	1.96e-05	5.46e-10	1.06e-66	9.47e-133	0	0	0	0
ADE-SA	1.76e-33	1.99e-62	8.82e-06	3.93e-10	9.21e-16	4.31e-30	0	0	0	0

Ave: mean value of the 50 independent trials. Std: standard deviation of the 50 independent trials.

TABLE 5: Hourly load demands of Case 1.

Time (h)	1	2	3	4	5	6	7	8	9	10	11	12
$P_{D,t}$	1036	1110	1258	1406	1480	1628	1702	1776	1924	2072	2146	2220
Time (h)	13	14	15	16	17	18	19	20	21	22	23	24
$P_{D,t}$	2072	1924	1776	1554	1480	1628	1776	2072	1924	1628	1332	1184

TABLE 6: Unit data of Case 1.

Time (h)	1	2	3	4	5	6	7	8	9	10
$a_i$ (\$/MW <sup>2</sup> h)	0.00043	0.00063	0.00039	0.0007	0.00079	0.00056	0.00211	0.0048	0.10908	0.00951
$b_i$ (\$/MWh)	21.6	21.05	20.81	23.9	21.62	17.87	16.51	23.23	19.58	22.54
$c_i$ (\$/h)	958.2	1313.6	604.97	471.6	480.29	601.75	502.7	639.4	455.6	692.4
$e_i$ (\$/h)	450	600	320	260	280	310	300	340	270	380
$f_i$ (rad/MW)	0.041	0.036	0.028	0.052	0.063	0.048	0.086	0.082	0.098	0.094
$P_{i,min}$ (MW)	150	135	73	60	73	57	20	47	20	55
$P_{i,max}$ (MW)	470	460	340	300	243	160	130	120	80	55
$DR_i$ (MW/h)	80	80	80	50	50	50	30	30	30	30
$UR_i$ (MW/h)	80	80	80	50	50	50	30	30	30	30

Case 4. 54-generator system.

In this case, a challenging problem with 54 generators is employed. The data and information of this system can be obtained from [35]. The simulation results of the proposed methods are compared in Table 14 with GA [15], PSO [15], ICA [15], HS [35], IHS [46], GHS [46] and NPAHS [46]. Note that even the maximum cost obtained by these four methods, which is 1,633,187 \$/day, is better than the minimum cost of the other methods.

Case 5 to 7. 100-generator, 200-generator, and 500-generator systems.

To test the performance of ADE-SA to deal with large-scale DED problems, we apply ADE-SA to three larger-scale DED problems in case 5 to case 7, which contain 100 generators, 200 generators, and 500 generators derived from the 10-generator system of case 1. Compared with previous cases, the increase of the number of process variables and constraints in Case 5 to Case 7 makes the DED problem much more complex. For Case 5, the simulation results obtained by ADE-SA are compared with GA [45], PSO [45], FA [45], FAPSO [47], SAFA [47], and CSO [43]. It can be seen from Table 15 that ADE-SA outperforms any other method regardless of minimum cost, average cost, and maximum

TABLE 7: Best result obtained by ADE-SA for Case 1 (10-unit system without  $P_L$ ).

Time (h)	1	2	3	4	5	6	7	8	9	10
1	150	222.26	156.69	60	73	122.45	129.59	47	20	55
2	150	223.56	229.4	60	73	122.45	129.59	47	20	55
3	150	303.56	297.4	60	73	122.45	129.59	47	20	55
4	226.62	316.8	305.67	60	122.87	122.45	129.59	47	20	55
5	226.62	396.8	299.67	60	122.87	122.45	129.59	47	20	55
6	303.25	396.8	321.17	60	172.73	122.45	129.59	47	20	55
7	303.25	396.8	297.4	107.91	222.6	122.45	129.59	47	20	55
8	379.87	396.8	302.14	120.42	172.73	122.45	129.59	77	20	55
9	456.5	396.8	297.4	136.43	222.59	122.45	129.59	87.24	20	55
10	456.5	396.8	297.83	186.43	222.6	160	129.59	117.24	50	55
11	456.5	396.8	316.1	236.43	223.52	160	129.59	120	52.06	55
12	456.5	460	330.95	233.3	222.6	160	129.59	120	52.06	55
13	456.5	396.8	297.85	183.3	222.6	160	129.59	120	50.36	55
14	456.5	396.8	297.4	133.3	222.6	122.45	129.59	90	20.36	55
15	379.87	396.8	312.55	110	172.74	122.45	129.59	77	20	55
16	303.25	396.8	297.05	60	122.86	122.45	129.59	47	20	55
17	226.62	396.8	299.67	60	122.86	122.45	129.59	47	20	55
18	303.25	396.8	291.18	60	172.73	122.45	129.59	77	20	55
19	379.87	396.79	297.4	66.98	222.59	122.45	129.59	85.31	20	55
20	456.49	460	336.02	116.98	222.6	160	129.59	85.31	50	55
21	456.5	389.53	297.4	110	222.6	158.07	129.59	85.31	20	55
22	379.87	309.53	301.83	60	172.73	122.45	129.59	77	20	55
23	303.25	229.53	242.31	60	122.87	122.45	129.59	47	20	55
24	226.56	222.23	180.35	60	120.8303297	122.44	129.59	47	20	55
Total cost (\$)	1,016,412.81									

TABLE 8: Statistical comparison for Case 1 (10-unit system without  $P_L$ ).

Method	$F(B)$ (\$)	$F(A)$ (\$)	$F(W)$ (\$)	$SD$	$Time$ (min)
SQP [11]	1,051,163	NA	NA	NA	0.421
GA [10]	1,033,481	1,038,014	1,042,606	NA	NA
MDE [22]	1,031,612	1,033,630	NA	NA	4.417
HDE [39]	1,031,077	NA	NA	NA	NA
PSO [10]	1,027,679	1,031,716	1,034,340	NA	NA
AIS [9]	1,021,980	1,023,156	1,024,973	NA	25.346
IDE [12]	1,026,269	NA	NA	NA	16.333
ABC [10]	1,021,576	1,022,686	1,024,316	NA	2.603
TLA [16]	1,019,925	1,020,411	1,021,118	NA	0.049
CDE [23]	1,019,123	1,020,870	1,023,115	1310.70	0.27
ICPSO [21]	1,019,072	1,020,027	NA	493.21	0.350
EAPSO [40]	1,018,510	1,018,710	1,019,302	NA	0.625
HIGA [41]	1,018,473	1,019,328	1,022,284	NA	NA
TVAC-IPSO [20]	1,018,217	1,018,965	1,020,417	NA	2.718
BBPSO [42]	1,018,159	1,019,850	1,021,813	826.94	NA
CSO [43]	1,017,660	1,018,120	1,019,286	302.3103	NA
EBSO [44]	1,017,147	1,017,526	1,017,891	147.01	0.025
MTLA [16]	1,016,935	1,016,972	1,017,091	NA	0.065
SAMFA [45]	1,016,808	1,017,030	1,017,187	NA	0.147
MIQP [31]	1,016,601	1,016,601	1,016,601	NA	1.884
ADE-SA	1,016,412	1,016,432	1,016,465	19.21	1.805

NA denotes that the value was not available in the literature.

TABLE 9: Best result obtained by ADE-SA for Case 1 (10-unit system with  $P_L$ ).

Time (h)	1	2	3	4	5	6	7	8	9	10	$P_L$
1	150	135	206.17	60	122.85	122.43	129.59	47	20	55	12.13
2	150	135	282.82	60	122.82	122.37	129.59	47	20	55	14.70
3	226.58	142.24	300.03	60	172.82	122.59	129.59	47	20	55	17.95
4	303.26	222.24	297.40	60	172.76	122.28	129.59	47	20	55	23.62
5	379.79	222.27	297.42	60	172.67	122.47	129.59	47	20	55	26.30
6	456.39	229.39	308.18	70.22	222.58	122.44	129.59	47	20	55	32.90
7	456.40	309.39	303.81	120.22	172.67	122.41	129.59	48.86	20	55	36.46
8	456.41	309.94	297.37	170.21	172.69	122.44	129.59	78.86	20	55	36.61
9	456.48	389.94	298.13	190.35	222.49	122.53	129.59	85.38	20	55	45.99
10	456.49	396.80	324.78	240.35	222.51	160.00	129.59	115.38	20	55	49.00
11	456.98	396.83	340	290.35	227.67	160.00	130.00	120.00	20.05	55	50.97
12	456.49	460.00	325	300	222.60	159.99	129.59	120.00	50.00	55	58.77
13	456.49	396.80	298.45	300	222.50	122.45	129.59	120.00	20	55	49.38
14	456.43	316.80	302.33	250	222.56	122.45	129.59	90.00	20	55	41.27
15	379.81	308.04	296.11	241.18	172.71	122.37	129.59	85.31	20	55	34.22
16	303.18	228.04	319.79	191.18	122.84	122.45	129.59	85.31	20	55	23.48
17	226.50	222.26	288.43	180.30	172.63	122.30	129.59	85.31	20	55	22.42
18	303.19	302.26	286.40	180.60	172.66	122.38	129.59	85.31	20	55	29.48
19	379.90	316.79	298.99	180.86	222.66	122.68	129.59	85.31	20	55	35.88
20	459.90	396.79	340.00	181.37	243.00	160.00	130.00	83.54	20	55	51.09
21	456.52	389.65	319.10	169.87	222.74	122.64	129.86	84.21	20	55	45.70
22	379.83	309.65	297.02	119.90	172.75	122.47	129.59	54.21	20	55	32.51
23	303.18	229.65	253.80	69.90	122.77	122.40	129.59	47	20	55	21.36
24	226.57	222.24	195.14	60	122.84	123.23	129.59	47	20	55	17.69
Total cost (\$)	1,037,140.18										

TABLE 10: Statistical comparison for Case 1 (10-unit system with  $P_L$ ).

Method	$F(B)$ (\$)	$F(A)$ (\$)	$F(W)$ (\$)	$SD$	$Time$ (min)
AIS [9]	1,045,715	1,047,050	1,048,431	NA	30.973
ABC [10]	1,043,381	1,044,963	1,046,805	NA	3.408
TVAC-IPSO [20]	1,041,066	1,042,118	1,043,625	NA	3.155
EBSO [44]	1,038,915	1,039,188	1,039,272	NA	0.22
MTLA [16]	1,037,489	1,037,712	1,038,090	NA	0.111
SAMFA [45]	1,037,698	1,037,938	1,039,199	NA	0.253
MIQP [31]	1,038,376	NA	NA	NA	1.977
ADE-SA	1,037,140	1,037,453	1,037,776	208.17	3.090

NA denotes that the value was not available in the literature.

cost. For Case 6 and Case 7 with 200 and 500 generators, only CSO reported results so far.

**5.3. Learning Cycle Study.** In the learning cycle, the performance of each mutation operators is evaluated and used to determine the chosen probability in the latter stage of search process. Figure 2 shows that five mutation operators are adapting to the whole evolution process. The chosen probability of each operator can be seen from the figure. In the initial of the evolution process, M1 and M2 showed high chosen probability whereas the other operators showed low probability. As the iteration proceeds, the chosen probability of M4 and M5 increased continuously.

**5.4. Adaptive Selection Mechanism Study.** To demonstrate the effect of the proposed adaptive selection mechanism, the average of best objective function values over 50 independent runs of each mutation operator are recorded in Table 16 (take Case 1, for example). From this table, it can be seen that each operator shows poor performance when runs depend on themselves. When the adaptive selection mechanism is applied, the operators are able to obtain high-quality solutions.

**5.5. Heuristic for Constraint Handling Study.** The proposed heuristic procedure can help the infeasible solutions to transform into high-quality feasible solutions. To test the

TABLE 11: Power generation output for Case 2.

Time (h)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	$P_L$
1	380.98	298.25	130	130	150.01	448.49	465	60	25	25	35.16	52.20	25	15	15	19.08
2	378.83	284.48	130	130	150	444.52	465	60	25	25	34.92	50.99	25	15	15	18.75
3	381.15	289.64	130	130	150	447.94	465	60	25	25	34.51	51.68	25	15	15	18.92
4	383.80	290.61	130	130	150	453.78	465	60	25	25	35.26	51.61	25	15	15	19.05
5	400.33	326.66	130	130	150	460	465	60	25	25	36.79	54.31	25	15	15	20.09
6	413.27	329.53	130	130	150	460	465	60	25	25	38.57	55.01	25	15	15	20.39
7	413.17	343.64	130	130	150	460	465	60	25	25	39.32	55.54	25	15	15	20.68
8	453.46	408.73	130	130	150	460	465	60	25	25	43.46	60.25	25	15	15	22.91
9	455	455	130	130	230.00	460	465	60	25	53.29	79.96	80	25	15	15	27.26
10	455	455	130	130	305.47	460	465	60	25	58.08	80	80	25	15	15	30.55
11	455	455	130	130	346.99	460	465	60	25	74.13	80	80	25	15	15	33.11
12	455	455	130	130	348.44	460	465	60	25	74.77	80	80	25	15	15	33.21
13	455	455	130	130	344.75	460	465	60	25	73.21	80	80	25	15	15	32.96
14	455	455	130	130	382.10	460	465	60	25	88.50	80	80	25	15	15	35.60
15	455	455	130	130	462.10	460	465	60	25	139.01	80	80	25	15	15	43.11
16	455	455	130	130	470	460	465	60	25	128.20	80	80	25	15	15	43.20
17	455	455	130	130	438.24	460	465	60	25	108.81	80	80	25	15	15	40.05
18	455	455	130	130	362.05	460	465	60	25	80.09	80	80	25	15	15	34.14
19	455	455	130	130	249.55	460	465	60	25	34.79	79.33	80	25	15	15	27.67
20	455	455	130	130	202.81	460	465	60	25	25.00	67.76	79.15	25	15	15	25.73
21	450.91	397.86	130	130	150	460	465	60	25	25.00	45.81	60.03	25	15	15	22.62
22	412.72	327.14	130	130	150.02	460	465	60	25	25.00	38.72	53.73	25	15	15	20.33
23	387.71	304.11	130	130	150	460	465	60	25	25.00	35.63	52.99	25	15	15	19.44
24	383.95	304.68	130	130	150	456.93	465	60	25	25.00	35.51	52.28	25	15	15	19.35
Total cost (\$)	759,020,42															

TABLE 12: Statistical comparison for Case 2 (15-unit system with  $P_L$ ).

Method	$F(B)$ (\$)	$F(A)$ (\$)	$F(W)$ (\$)	$SD$	$Time$ (min)
HS [46]	765,560	765,959	766,370	NA	11.305
HIS [46]	765,600	765,942	766,403	NA	11.358
GHS [46]	769,074	769,627	770,428	NA	32.252
SGHS [46]	759,897	760,118	760,343	NA	5.055
NPAHS [46]	759,603	759,779	759,988	NA	4.167
ODO-ABC	759,020	759,247	759,481	77.86	3.931

NA denotes that the value was not available in the literature.

TABLE 13: Statistical comparison for Case 3 (30-unit system derived from 10-unit system).

Method	$F(B)$ (\$)	$F(A)$ (\$)	$F(W)$ (\$)	$SD$	$Time$ (min)
DE [23]	3,163,000	3,173,100	NA	NA	NA
CDE [23]	3,083,930	3,090,542	NA	NA	0.58
IPSO [18]	3,090,570	NA	3,096,900	NA	0.142
CSAPSO [47]	3,066,907	3,075,023	NA	NA	0.765
ICPSO [21]	3,064,497	NA	3,071,588	NA	0.773
EAPSO [40]	3,054,961	3,055,641	3,055,257	NA	NA
BBPSO [42]	3,062,144	3,067,277	NA	NA	NA
MTLA [16]	3,048,609	3,051,113	3,049,871	NA	0.127
CSADHS [13]	3,054,709	3,055,137.84	3,055,070	NA	NA
DGPSO [19]	3,148,992	3,154,438	NA	2177.60 6.3	22.816
ADE-SA	3,047,318	3,047,478	3,047,542	115.74	2.175

NA denotes that the value was not available in the literature.

TABLE 14: Statistical comparison for Case 4 (54-unit system).

Method	$F(B)$ (\$)	$F(A)$ (\$)	$F(W)$ (\$)	$SD$	$Time$ (min)
GA [15]	1,834,373	1,839,422	1,850,775	NA	NA
PSO [15]	1,832,121	1,835,851	1,845,937	NA	NA
ICA [15]	1,807,081	1,809,664	1,811,388	NA	NA
HS [46]	1,764,951	1,767,045	1,770,005	NA	5.06
IHS [46]	1,763,261	1,766,792	1,771,102	NA	4.598
GHS [46]	1,801,693	1,804,643	1,809,412	NA	4.595
NPAHS [46]	1,710,489	1,712,278	1,715,3	NA	4.433
OCD [48]	1,772,724	NA	NA	NA	0.0022
ADE-SA	1,632,628	1,652,744	1,653,187	1202.6	4.126

NA denotes that the value was not available in the literature.

TABLE 15: Statistical comparison for Case 5-7.

Method	$F(B)$ (\$)	$F(A)$ (\$)	$F(W)$ (\$)	$SD$	$Time$ (min)
<b>Case 5: 100-unit system</b>					
GA [45]	10,908,741	11,584,628	11,987,675	NA	9.04
PSO [45]	10,366,076	10,766,385	11,310,279	NA	5.89
FA [45]	10,197,269	10,419,457	11,216,243	NA	1.3
FAPSO [47]	10,200,860	10,207,245	10,211,461	NA	NA
SAFA [45]	10,183,819	10,286,043	10,388,958	NA	1.41
CSO [43]	10,183,633	10,185,287	10,192,352	1323	NA
ADE-SA	10,181,267	10,181,953	10,182,649	749.47	7.692
<b>Case 6: 200-unit system</b>					
CSO [43]	20,409,722	20,414,842	20,437,874	4792	NA
ADE-SA	20,392,355	20,398,717	20,405,289	2017.10	13.542
<b>Case 7: 500-unit system</b>					
CSO [43]	51,044,611	51,050,457	51,082,986	6067	NA
ADE-SA	51,032,830	51,040,129	51,055,302	4109.83	31.169

NA denotes that the value was not available in the literature.

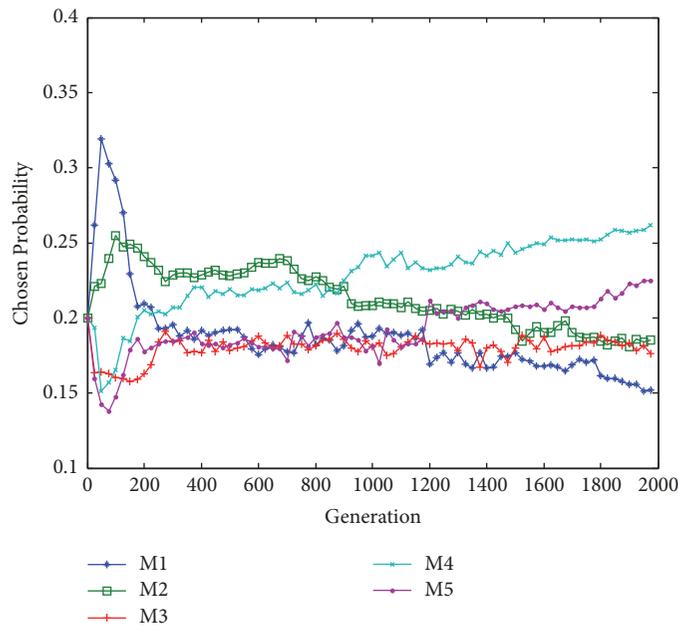


FIGURE 2: Chosen probability of mutation operators.

TABLE 16: Comparison of average solutions obtained by M1-M5 and ADE-SA.

Method	$F(B)$ (\$)	$F(A)$ (\$)	$F(W)$ (\$)	$SD$
M1	1,023,027	1,023,288	1,024,356	240.61
M2	1,017,893	1,018,363	1,019,012	523.10
M3	1,019,240	1,019,754	1,020,667	741.74
M4	1,020,533	1,020,972	1,021,702	651.83
M5	1,018,315	1,018,646	1,019,384	508.82
ADE-SA	1,016,412	1,016,432	1,016,465	19.21

TABLE 17: Comparison with and without the heuristic procedure.

Method	Heuristic procedure	Mean cost (\$)	Failure times
M1	Without	1,025,109	55,453
	With	1,023,288	34,576
M2	Without	1,019,392	61,929
	With	1,018,363	44,531
M3	Without	1,020,923	42,106
	With	1,019,754	22,374
M4	Without	1,022,165	40,827
	With	1,020,972	23,201
M5	Without	1,020,131	47,233
	With	1,018,646	29,809
ADE-SA	Without	1,017,254	50,142
	With	1,016,432	32,849

effect of this heuristic procedure, we record the average cost over 20 independent runs and the failure times of constraint handling with and without the heuristic procedure in the same experiment situation. The penalty method, which is one of the most frequent methods for dealing with DED problem, is used for comparison. Table 17 shows that each operator with the heuristic procedure can find better solution. In the constraint handling process, the infeasible solution is adjusted hour by hour. When a feasible solution  $P_t$  is obtained at current hour, it may make it impossible for  $P_{t+1}$  to meet the power balance and ramp rate constraints at the next hour. Failure times counts the times of the situation. A lower failure times means an effective constraint handling strategy. All the experiments have the same 2000 generations.

## 6. Conclusions

This paper proposes and successfully applies a novel method to solve the DED problem considering value-point effects. DE is employed to search the feasible solution of DED. Five widely used mutation operators are gathered to form a mutation operator pool. Simulated annealing algorithm is used to carry out an adaptive selection mechanism in which the mutation operators of DE are selected adaptively. The effectiveness of the proposed methods is demonstrated first on 10 popular benchmark functions with 100 dimensions, in comparison with the classic DE and five variants. Then, seven test systems having 10, 15, 30, 54, 100, 200, and 500 generators are analyzed to illustrate the application of the proposed method. The computational results show that the proposed

method has the capacity to provide stable and high-quality solutions efficiently for DED problems. Comparison of the results with algorithms recently reported in the literature confirms the superiority of the proposed method and its higher probability of finding accurate and optimal solutions for DED problems. Finally, the performance of different components is discussed.

In future work, we plan to extend the present work in the following directions. First, adaptive techniques for parameter setting of the mutation operator will be investigated. Second, we will apply the proposed method to more complex practical DED problems that consider the uncertainty of load demand, wind speed, and solar irradiation.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants nos. 61773105, 61374147, 61703085, 61004083, and 61533007 and in part by the Fundamental Research Funds for the Central Universities under Grants nos. N150402001 and N120404014.

## References

- [1] X. Xia and A. M. Elaiw, "Optimal dynamic economic dispatch of generation: a review," *Electric Power Systems Research*, vol. 80, no. 8, pp. 975–986, 2010.
- [2] A. Theerthamalai and S. Maheswarapu, "An effective non-iterative " $\lambda$ -logic based" algorithm for economic dispatch of generators with cubic fuel cost function," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 5, pp. 539–542, 2010.
- [3] D. Srinivasan, F. Wen, C. S. Chang, and A. C. Liew, "A survey of applications of evolutionary computing to power systems," in *Proceedings of the Conference on Intelligent Systems Applications To Power Systems, 1996.*, *Proceedings*, pp. 35–41, 2002.
- [4] S. Hemamalini and S. P. Simon, "Dynamic economic dispatch using Maclaurin series based Lagrangian method," *Energy Conversion and Management*, vol. 51, no. 11, pp. 2212–2219, 2010.
- [5] D. L. Travers and R. John Kaye, "Dynamic dispatch by constructive dynamic programming," *IEEE Transactions on Power Systems*, vol. 13, no. 1, pp. 72–78, 1998.
- [6] J. C. Lee, W. M. Lin, G. C. Liao, and T. P. Tsao, "Quantum genetic algorithm for dynamic economic dispatch with valve-point effects and including wind power system," in *Proceedings of the International Journal of Electrical Power Energy Systems*, vol. 33, pp. 189–197, 2011.
- [7] B. K. Panigrahi, V. Ravikumar Pandi, and S. Das, "Adaptive particle swarm optimization approach for static and dynamic economic load dispatch," *Energy Conversion and Management*, vol. 49, no. 6, pp. 1407–1415, 2008.
- [8] C. K. Panigrahi, P. K. Chattopadhyay, R. N. Chakrabarti, and M. Basu, "Simulated annealing technique for dynamic economic dispatch," *Electric Power Components and Systems*, vol. 34, no. 5, pp. 577–586, 2006.
- [9] S. Hemamalini and S. P. Simon, "Dynamic economic dispatch using artificial immune system for units with valve-point effect," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 4, pp. 868–874, 2011.
- [10] S. Hemamalini and S. P. Simon, "Dynamic economic dispatch using artificial bee colony algorithm for units with valve-point effect," *European Transactions on Electrical Power*, vol. 21, no. 1, pp. 70–81, 2011.
- [11] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, "A hybrid ep and sqp for dynamic economic dispatch with nonsmooth fuel cost function," *Power Engineering Review, IEEE*, vol. 22, no. 4, pp. 77–77, 2007.
- [12] G. Marsala, G. Cirrincione, M. Pucci, and M. Cirrincione, "The GMR Neural networks for inverse problems," *Journal of Electrical Systems*, vol. 3, no. 3, pp. 176–188, 2007.
- [13] R. Arul, G. Ravi, and S. Velusami, "Chaotic self-adaptive differential harmony search algorithm based dynamic economic dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 50, no. 1, pp. 85–96, 2013.
- [14] S. Sivasubramani and K. S. Swarup, "Hybrid SOA-SQP algorithm for dynamic economic dispatch with valve-point effects," *Energy*, vol. 35, no. 12, pp. 5031–5036, 2010.
- [15] B. Mohammadi-Ivatloo, A. Rabiee, A. Soroudi, and M. Ehsan, "Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch," *Energy*, vol. 44, no. 1, pp. 228–240, 2012.
- [16] T. Niknam, R. Azizpanah-Abarghoee, and J. Aghaei, "A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 749–763, 2013.
- [17] R. Vijay, "Intelligent bacterial foraging optimization technique to economic load dispatch problem," *International Journal of Soft Computing & Engineering*, vol. 2, 2012.
- [18] X. Yuan, A. Su, Y. Yuan, H. Nie, and L. Wang, "An improved PSO for dynamic load dispatch of generators with valve-point effects," *Energy*, vol. 34, no. 1, pp. 67–74, 2009.
- [19] T. A. A. Victoire and A. E. Jeyakumar, "Deterministically guided PSO for dynamic dispatch considering valve-point effect," *Electric Power Systems Research*, vol. 73, no. 3, pp. 313–322, 2005.
- [20] B. Mohammadi-Ivatloo, A. Rabiee, and M. Ehsan, "Time-varying acceleration coefficients IPSO for solving dynamic economic dispatch with non-smooth cost function," *Energy Conversion and Management*, vol. 56, pp. 175–183, 2012.
- [21] Y. Wang, J. Zhou, Y. Lu, and H. Qin, "Chaotic self-adaptive particle swarm optimization algorithm for dynamic economic dispatch problem with valve-point effects," *Energy Conversion & Management*, vol. 38, no. 11, pp. 14231–14237, 2011.
- [22] X. H. Yuan, L. Wang, Y. B. Yuan, Y. Zhang, B. Cao, and B. Yang, "A modified differential evolution approach for dynamic economic dispatch with valve-point effects," *Energy Conversion and Management*, vol. 49, no. 12, pp. 3447–3453, 2008.
- [23] Y. Lu, J. Zhou, H. Qin, Y. Wang, and Y. Zhang, "Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 2, pp. 378–387, 2011.
- [24] D. He, G. Dong, F.L. Wang, and Z. Mao, "Optimization of dynamic economic dispatch with valve-point effect using chaotic sequence based differential evolution algorithms," *Energy Conversion and Management*, vol. 52, no. 2, pp. 1026–1032, 2011.
- [25] D. C. Secui, "A new modified artificial bee colony algorithm for the economic dispatch problem," *Energy Conversion and Management*, vol. 89, pp. 43–62, 2015.
- [26] P. K. Singhal, R. Naresh, and V. Sharma, "A modified binary artificial bee colony algorithm for ramp rate constrained unit commitment problem," *International Transactions on Electrical Energy Systems*, vol. 25, no. 12, pp. 3472–3491, 2015.
- [27] A. N. Afandi and H. Miyauchi, "Improved artificial bee colony algorithm considering harvest season for computing economic dispatch on power system," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 9, no. 3, pp. 251–257, 2014.
- [28] D. He, F. Wang, and Z. Mao, "A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect," in *Proceedings of the International Journal of Electrical Power Energy Systems*, vol. 30, pp. 31–38, 2008.
- [29] T. A. A. Victoire and A. E. Jeyakumar, "Reserve constrained dynamic dispatch of units with valve-point effects," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1273–1282, 2005.
- [30] M. Basu, "Hybridization of bee colony optimization and sequential quadratic programming for dynamic economic dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 44, no. 1, pp. 591–596, 2013.
- [31] M. Q. Wang, H. B. Gooi, S. X. Chen, and S. Lu, "A mixed integer quadratic programming for dynamic economic dispatch with valve point effect," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2097–2106, 2014.
- [32] K. Price, *Differential Evolution A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*, Kluwer Academic Publishers, 1997.

- [33] T. Sousa, A. Silva, and A. Neves, "Particle swarm based data mining algorithms for classification tasks," *Parallel Computing*, vol. 30, no. 5-6, pp. 767-783, 2004.
- [34] K. Veeramachaneni and L. A. Osadciw, Optimal scheduling in sensor networks using swarm intelligence, 2004.
- [35] B. Yu, X. Yuan, and J. Wang, "Short-term hydro-thermal scheduling using particle swarm optimization method," *Energy Conversion and Management*, vol. 48, no. 7, pp. 1902-1908, 2007.
- [36] C. O. Ourique, E. C. Biscaia Jr., and J. C. Pinto, "The use of particle swarm optimization for dynamical analysis in chemical processes," *Computers & Chemical Engineering*, vol. 26, no. 12, pp. 1783-1793, 2002.
- [37] J. Louchet, M. Guyon, M.-J. Lesot, and A. Boumaza, "Dynamic flies: A new pattern recognition tool applied to stereo sequence processing," *Pattern Recognition Letters*, vol. 23, no. 1-3, pp. 335-345, 2002.
- [38] D. He, L. Yang, Z. Wang, and X. Tian, *An Overlapped Decomposition Optimization Method for Dynamic Economic Dispatch*, IEEE ACCESS, 2018.
- [39] X. Yuan, L. Wang, Y. Zhang, and Y. Yuan, "A hybrid differential evolution method for dynamic economic dispatch with valve-point effects," *Expert Systems with Applications*, vol. 36, no. 2, pp. 4042-4048, 2009.
- [40] T. Niknam and F. Golestaneh, "Enhanced adaptive particle swarm optimization algorithm for dynamic economic dispatch of units considering valve-point effects and ramp rates," *IET Generation, Transmission & Distribution*, vol. 6, no. 5, pp. 424-435, 2012.
- [41] B. Mohammadi-Ivatloo, A. Rabiee, and A. Soroudi, "Nonconvex dynamic economic power dispatch problems solution using hybrid immune-genetic algorithm," *IEEE Systems Journal*, vol. 7, no. 4, pp. 777-785, 2013.
- [42] Y. Zhang, D.-W. Gong, N. Geng, and X.-Y. Sun, "Hybrid bare-bones PSO for dynamic economic dispatch with valve-point effects," *Applied Soft Computing*, vol. 18, pp. 248-260, 2014.
- [43] A. Meng, H. Hu, H. Yin, X. Peng, and Z. Guo, "Crisscross optimization algorithm for large-scale dynamic economic dispatch problem with valve-point effects," *Energy*, vol. 93, pp. 2175-2190, 2015.
- [44] T. Niknam and F. Golestaneh, "Enhanced bee swarm optimization algorithm for dynamic economic dispatch," *IEEE Systems Journal*, vol. 7, no. 4, pp. 754-762, 2013.
- [45] T. Niknam, R. Azizipanah-Abarghooee, and A. Roosta, "Reserve constrained dynamic economic dispatch: A new fast self-adaptive modified firefly algorithm," *IEEE Systems Journal*, vol. 6, no. 4, pp. 635-646, 2012.
- [46] Q. Niu, H. Zhang, K. Li, and G. W. Irwin, "An efficient harmony search with new pitch adjustment for dynamic economic dispatch," *Energy*, vol. 65, pp. 25-43, 2014.
- [47] J. Aghaei, T. Niknam, R. Azizipanah-Abarghooee, and J. M. Arroyo, "Scenario-based dynamic economic emission dispatch considering load and wind power uncertainties," *International Journal of Electrical Power & Energy Systems*, vol. 47, no. 1, pp. 351-367, 2013.
- [48] A. Rabiee, B. Mohammadi-Ivatloo, and M. Moradi-Dalvand, "Fast dynamic economic power dispatch problems solution via optimality condition decomposition," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 982-983, 2014.
- [49] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087-1092, 1953.
- [50] 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.
- [51] R. Bai, J. Blazewicz, E. K. Burke, G. Kendall, and B. McCollum, "A simulated annealing hyper-heuristic methodology for flexible decision support," *4OR*, vol. 10, no. 1, pp. 43-66, 2012.
- [52] S. Wang, Y. Li, and H. Yang, "Self-adaptive differential evolution algorithm with improved mutation mode," *Applied Intelligence*, vol. 47, no. 3, pp. 644-658, 2017.
- [53] M. Singh and J. S. Dhillon, "A Simple Opposition-based Greedy Heuristic Search for Dynamic Economic Thermal Power Dispatch," *Electric Power Components and Systems*, vol. 44, no. 6, pp. 589-605, 2016.

