Hindawi
Complexity
Volume 2022, Article ID 7275088, 18 pages
https://doi.org/10.1155/2022/7275088

Research Article

Differential Evolution with Autonomous Selection of Mutation Strategies and Control Parameters and Its Application

Zhenyu Wang,[1] Zijian Cao,[1] Zhiqiang Du,[1] Haowen Jia,[1] Binhui Han,[2] Feng Tian,[3] and Fuxi Liu[4]

[1] *School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, China*
[2] *School of Automotive Engineering, Xi'an Aeronautical Polytechnic Institute, Xi'an 710089, China*
[3] *Kunshan Duke University, Suzhou, China*
[4] *School of Mechanical and Electrical Engineering, Hunan Applied Technology University, Changde 415100, China*

Correspondence should be addressed to Fuxi Liu; liufx28@163.com

Received 14 December 2021; Revised 11 April 2022; Accepted 15 June 2022; Published 21 July 2022

Academic Editor: Alex Alexandridis

The existing numerous adaptive variants of differential evolution (DE) have been improved the search ability of classic DE to certain extent. Nevertheless, those variants of DE do not obtain the promising performance in solving black box problems with unknown features, which is mainly because the adaptive rules of those variants are designed according to their designers' cognition on the problem features. To enhance the optimization ability of DE in optimizing black box problems with unknown features, a differential evolution with autonomous selection of mutation strategies and control parameters (ASDE) is proposed in this paper, inspired by autonomous decision-making mechanism of reinforcement learning. In ASDE, a historical experience archive with population features is utilized to preserve accumulated historical experience of the combination of mutation strategies and control parameters. Furthermore, the accumulated historical experience can be autonomously mapped into rules repository, and the individuals can choose the combination of mutation strategies and control parameters according to those rules. Additionally, an updating and utilization mechanism of the historical experience is designed to assure that the historical experience can be effectively accumulated and utilized efficiently. Compared with some state-of-the-art intelligence algorithms on 15 functions of CEC2015, 28 functions of CEC2017, and parameter extraction problems of the photovoltaic model, ASDE has the advantages of solution accuracy, convergence speed, and robustness in solving black box problems with unknown features.

1. Introduction

Differential evolution (DE) algorithm is an effective and efficient global search engine for complex optimization problems, to solve the Chebyshev polynomial problem at first proposed by Storn and Price [1]. Since DE is a simple and robust optimizer, DE has been a hot topic in the field of intelligent optimization algorithm. In DE, a new solution of the problem is generated by using the scaled difference vector between two distinct solution randomly selected from the candidate solution set. Meanwhile, the one-to-one selection strategy is utilized to choose the better solution between parent and offspring individual to propagate to next generation. In addition, the implementation of DE only needs a few lines of code in any standard programming language, which makes it easy to realize for engineers of different optimization fields. Over the past two decades, DE and other intelligent algorithms have gained the promising performance in solving numerous practical engineering problems, such as, chemical engineering [2–5], electrical engineering [6–9], scheduling optimization [10, 11], image processing [12–15], and structural optimization of neural network [16, 17]. The convergence of DE and its variants is proved in the literature [18].

The performance of DE is significantly influenced by mutation strategies and control parameters, and numerous research has been done for finding out the proper mutation strategies or/and control parameters.

In terms of adjusting control parameters, in the literature [19], F is in between 0.4 and 0.95, and the initialized F is 0.9. Moreover, CR should be between 0 and 0.2 when the objective problem is separable, and [0.9, 1] otherwise. To improve the performance of DE, Mohamed A. K. and Mohamed A. W. [20] proposed an enhanced AGDE algorithm, named EAGDE for short. In EAGDE, population size gradually decreases according to a nonlinear population size reduction.

In terms of designing mutation strategies, Cui et al. [21] proposed an adaptive multiple-elites-guided composite DE with a shift mechanism, and the best one of two trial vector generated by two elites-guided trial vector generation strategies, respectively, is adopted to participate in the selection. In addition, a shift mechanism is used to avoid falling into local trap. Mohamed A. W. and Mohamed A. K. [22] constructed a novel mutation strategy, which uses two random selected vectors of the top and the bottom 100p% individuals of the population while the third vector is chosen randomly from the middle individuals.

However, the suitable mutation strategies and control parameters are problem specific. To assign appropriate mutation strategies or/and control parameters to the population, numerous adaptive variants were proposed to further improve the search ability of DE. These adaptive variants can be divided into adaptive control parameters, adaptive mutation strategies, and adaptive topological neighborhood.

In terms of adaptive control parameters, Zhao et al. [23] proposed a DE with self-adaptive control parameters and strategy for unconstrained optimization problems (SLADE). The Cauchy distribution and normal distribution were utilized to adaptively update the mutation factor F and crossover rate CR. Ghosh et al. [24] designed a very simple and flexible technology for adjusting F and CR online, and the adaptive method was based on the objection function values of different individuals in population. A new parameter with adaptive learning mechanism DE was proposed by Meng et al. [25] for solving the inconvenience selection problem of control parameter. Brest et al. [26] proposed the DE with autoadaptive control parameters for solving stagnation in local optima. A self-adapting control parameter in DE was proposed by Brest et al. [27], in which the control parameters were adjusted by means of evolution. A variant of DE (JADE) was proposed by Zhang and Sanderson [28], in which the values of F and CR are sampled from the Cauchy distribution and Gaussian distribution, respectively. In JADE, the control parameters of probability distribution are updated by adaptive strategy.

Tanabe and Fukunaga [29] proposed a success history based DE (SHADE), in which the historical memory archive is utilized to adaptive update the control parameters of probability distribution, instead of gradually update mechanism of JADE. Zhou et al. [10] proposed a new DE algorithm, in which control parameter values are adaptively determined according to their historical performances.

Regarding adaptive mutation strategies, in the above literature [23], the mutation strategy assigned to each individual is adaptively chose from the candidate strategy pool for matching different stage of evolution search according to their previous successful experience. Yu et al. [30] proposed a novel mutation DE for global optimization, in which the adaptive mutation is carried out for the current individual when individuals cluster around the local optimal solution. Mallipeddi and Suganthan [31] proposed a DE algorithm with ensemble of parameters and mutation and crossover strategies (EPSDE). In EPSDE, a set of different mutation strategies and the values of each control parameter coexist throughout the evolution search and compete to produce offspring solutions. In the literature [32], a new DE algorithm was proposed, which divides the population into three subpopulations according to the fitness value and applies three mutation strategies to exploration or exploitation. Qin et al. [33] proposed an adaptive DE algorithm (SaDE), in which several mutation strategies with different features are put into a candidate pool. After a certain interval period, a mutation strategy is adaptively selected according to its success rate. Wu et al. [34] proposed a DE with multi-population-based ensemble of mutation strategies (MPEDE). In MPEDE, three mutation strategies are allocated to three subpopulations for evolution search. After a period, the current best mutation strategy will be adaptively determined according to the improvement of fitness value, which will be assigned to the subpopulation with largest number in next search period.

Wu et al. [35] designed a novel DE variant by using ensemble of multiple DE variants, named EDEV for short. In EDEV, the population is partitioned into four subpopulations, including three indicator subpopulations with smaller size and one reward subpopulation with much larger size. Each constituent DE variant of EDEV owns an indicator subpopulation. The reward subpopulation is assigned to the DE variant with the best performance in previous generations.

About adaptive topological neighborhood, Das et al. [36] proposed a DE using a neighborhood-based mutation operator (DEGL). In DEGL, the concept of small neighborhood is defined on the index-graph of individuals, and the weight factor is used to dynamically adjust the mutation operator based on local model and global model, which effectively balances the exploration and exploitation ability. Cai and Wang [37] proposed a DE with neighborhood and direction information (NGDE), which utilizes a neighborhood selection mechanism based on population position order to guide individuals to search in a good direction. Epitropakis et al. [38] proposed enhancing DE utilizing proximity-based mutation operators (ProDE), in which each individual constructs a topological neighborhood according to the proximity index and adaptively selects the search direction generated by the individual that is closed to it. Wang et al. [39, 40] proposed two DE algorithms based on eigenvector crossover operation, which use the fitness value to construct the topological neighborhood of the population. Each individual of the population searches adaptively feasible solution space by learning the data distribution features of the population topological neighborhood.

Additionally, to alleviate the long time-consuming problem of the algorithm in fitness evaluation, Zhan et al.

[41] proposed a double-layered heterogeneous DE algorithm, in which different populations with various parameters or/and mutation strategies run concurrently and adaptively migrate to deliver robust solutions by making the best use of performance differences among multiple populations, and a set of cloud virtual machines run in parallel to evaluate fitness of corresponding populations, reducing computational costs as offered by cloud. In the literature [42], an adaptive distributed differential evolution is proposed for decreasing the sensitivity of strategies and parameters, named ADDE for short. In ADDE, three populations called exploration population, exploitation population, and balance population are co-evolved concurrently by using the master-slave multipopulation distributed framework. Different populations adaptively choose suitable mutation strategies according to their previous performance.

Although these adaptive methods have improved the performance of DE in solving nondifferentiable, nonlinear, and nonseparable problems to a certain extent, they do not address black box problems with unknown features because their adaptive rules are designed according to designers' cognition of the problem features. For example, in the literature [30], the current best individual will be adjusted adaptively when the population falls into a local optimal trap. To determine whether the population converges prematurely, a parameter d is designed to reflect the convergence degree of population. The population is considered to fall into the neighborhood of local optimal solution when the parameter d meets the predefined threshold that is set according to the designer's cognition of solving problems. When the population falls into the local optimal trap, the mutation strategy with good exploration ability will be assigned to the population for helping the population jump out the local optimal trap. Thus, the predefined threshold determines whether the population falls into local optimal trap, which determines the next search behavior of the population. In the literature [43], the whole process of population search is divided into five search stages in chronological order, and the mutation strategy is assigned to population according to the search stage features determined by designer.

Therefore, to optimize the black box problems with unknown features, a DE with autonomous selection of mutation strategies and control parameters (ASDE) is proposed in this paper. In ASDE, a historical experience archive with population features is utilized to store accumulated historical experience of the combination of mutation strategies and control parameters, and the accumulated historical experience can be autonomously mapped into rules repository so that the population can choose the appropriate combination of mutation strategies and control parameters for evolution search according to those rules. Furthermore, each individual of the population chooses a suitable combination of mutation strategies and control parameters to explore feasible solution space according to the historical experience whose population features is most similar to the population features of current population, and the improvement of fitness value of the individuals is used to update corresponding historical experience.

The main contributions of this paper can be summarized as follows:

(1) The historical experience archive with population features is proposed for storing cumulated historical experience so that individuals can autonomously select appropriate combination of mutation strategies and control parameters according to the rule mapped from the cumulated historical experience.

(2) The updating and utilization mechanism of the historical experience can assure that the historical experience of mutation strategies and control parameters can be effectively accumulated and utilized efficiently.

The rest of this paper is provided as follows. The brief review of canonical DE is described in Section 2. In Section 3, DE with autonomous selection of mutation strategies and control parameters is addressed in detail. Section 4 presents the experiment results on 15 functions of CEC2015. The results of parameter extraction of photovoltaic models are shown in Section 5, followed by conclusion in Section 6.

2. The Reviews of DE

The implement of canonical DE, proposed by Storn and Price [1], is composed of initialization, mutation, crossover, and selection, which are addressed briefly as follows.

2.1. Initialization. The initial population of DE is consisted of a group feasible solutions randomly selected form objective solution space, termed $pop^1 = \{X_i^1 = (x_{i,1}^1, x_{i,2}^1, \ldots, x_{i,D}^1) | i = 1, 2, \ldots, NP\}$, where NP denotes the population size and D represents the dimensions of problem. X_i is i th individual of population, which is generated by

$$X_i = lb + (ub - lb) * r\text{and}, \tag{1}$$

where lb and ub are lower and upper bounds of feasible solution space, respectively. $r$and is a random number, sampled uniformly and randomly from range [0, 1].

2.2. Mutation. The mutation operator simulates the gene mutation in nature, which is used to create a mutation vector $V_i^t = \{v_{i,1}^t, v_{i,2}^t, \ldots, v_{i,D}^t\}$ for each target vector at t th generation. V_i^t is calculated by

$$V_i^t = X_{r1}^t + F * \left(X_{r2}^t - X_{r3}^t\right), \tag{2}$$

where F is scaling factor, and value of F is into interval [0, 1]. $r1, r2, r3 \in \{1, 2, \ldots, NP\}$ are distinct indices.

2.3. Crossover. The mutation vector V_i^t is used to generate trial vector $U_i^t = \{u_{i,1}^t, u_{i,2}^t, \ldots, u_{i,D}^t\}$ by performing crossover addressed as

std_{f1}	std_{f2}	std_{f3}	$\ldots\ldots$	$std_{f(H\text{-}1)}$	std_{fH}
std_{pop1}	std_{pop2}	std_{pop3}	$\ldots\ldots$	$std_{pop(H\text{-}1)}$	std_{popH}
$stag_1$	$stag_2$	$stag_3$	$\ldots\ldots$	$stag_{(H\text{-}1)}$	$stag_H$
E_1	E_2	E_3	$\ldots\ldots$	$E_{(H\text{-}1)}$	E_H

Figure 1: The historical experience archive.

$$u_{i,j}^{t} = \begin{cases} x_{i,j}^{t}, & r\text{ and} < CR, \\ v_{i,j}^{t}, & \text{otherwise}, \end{cases} \tag{3}$$

where $x_{i,j}^{t}$ denotes the j th dimension data of i th individual and j is one of the sets $\{1, 2, \ldots, D\}$. In addition, CR represents the crossover rate and is into the interval [0, 1].

2.4. Selection. To make better individuals in the trial vectors and target vectors inherit to next generation, the selection is performed as formula (4), and minimization problems are considered in this paper:

$$X_{i}^{t+1} = \begin{cases} X_{i}^{t}, & if\ f\left(U_{i}^{t}\right) > f\left(X_{i}^{t}\right), \\ U_{i}^{t}, & \ldots\text{otherwise}, \end{cases} \tag{4}$$

where $f\,(X_{i}^{t})$ denotes calculating the fitness value of vector X_{i}^{t}.

Mutation, crossover, and selection will be repeated until the predefined termination criterion is met.

3. The Proposed Algorithm

In this section, ASDE will be detailly described in terms of historical experience archive, historical experience update and utilization, used mutation strategies and control parameters, and the architecture of proposed algorithm.

3.1. The Historical Experience Archive. It can be clearly shown from Figure 1 that the historical experience archive with H entries is consisted of population features (std_f, std_{pop}, and stag) and historical experience E, where std_f denotes the standard deviation of fitness value of the population, and std_{pop} are the sum of standard deviation of each dimension of individuals. The std_f and std_{pop} are calculated by formulas (5) and (6), respectively.

Meanwhile, the stag stands for the generation number of continuous stagnations of the population, updated according to formula (7), where the stagnation is referred to that the optimum solution is not improved during population evolution:

$$\text{std}_{f} = \sqrt{\sum_{i=1}^{NP} \left(f_{i} - \overline{f}\right)^{2}}, \tag{5}$$

$$\text{std}_{\text{pop}} = \sum_{j=1}^{D} \sqrt{\sum_{i=1}^{NP} \left(x_{i,j} - \overline{x_{i,j}}\right)^{2}}, \tag{6}$$

$$\text{stag} = \begin{cases} \text{stag} + 1, & \text{if optima isnt, improved}, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $\overline{f}$ is the mean value of fitness of population, and $\overline{x_{i,j}}$ denotes the mean value of j th dimension data of all individuals.

The std_f can indicate the difference between fitness values of all individuals in the population, and the std_{pop} can show the diversity properties of the population. The stag can indicate whether the population falls into the local optimal trap. Thus, the combination of std_f, std_{pop}, and stag can ideally describe the state of the current population.

In addition, the historical experience E is the special vector that is shown in Figure 2 and composed of the cumulated reward of m combinations of mutation strategies and control parameters, where the reward is the average nonnegative improvement of fitness value of the individuals that use corresponding combinations for evolution search.

3.2. The Updating and Utilization Mechanism of the Historical Experience. To assign most appropriate historical experience to current population, the updating and utilization mechanism of the historical experience is utilized, so that each individual of the current population can use proper combination of mutation strategies and control parameters to search according to the rules mapped from assigned historical experience. A population feature similarity calculation operator is mostly main operator of this mechanism, used to calculate the population feature similarity PFS between population features $PF^{t} = (\text{std}_f$, std_{pop} and stag) of current population and population features PF_k ($k \in \{1, 2, \ldots, H\}$) of the historical experience archive. The population feature similarity calculation operator is

R_{c1}	R_{c2}	……	R_{cm}

Figure 2: The historical experience.

described as formula (8), in which the combination of Mahalanobis distance and Euclidean distance is viewed as the population feature calculator. The Mahalanobis distance is an effective method to calculate the similarity of two unknown sample sets, because it is not affected by unit of each dimension data so that each dimensional data of the records of the historical experience archive can be processed equally. However, the Mahalanobis distance has a necessary condition that the covariance matrix of population feature sets of the historical experience archive must exist. Thus, a combination of Mahalanobis distance and Euclidean distance is used to calculate the population feature similarity:

$$PFS_k = \begin{cases} \text{Mahalanobis}\left(PF^t, PF_k\right), & \text{if the covariance matrix exists,} \\ \sqrt{\left(\text{std}_f - \text{std}_{fk}\right)^2 + \left(\text{std}_{pop} - \text{std}_{popk}\right)^2 + \left(\text{stag} - \text{stag}_k\right)^2}, & \text{otherwise,} \end{cases} \tag{8}$$

where Mahalanobis ($*$) denotes the Mahalanobis function that is utilized to calculate the Mahalanobis distance between PF^t and PF_k.

After the population feature similarity calculation operator is executed, the historical experience (E^t) with highest population feature similarity (the smaller the value of PFS_k, the higher the population feature similarity) is assigned to the current population. The ε-greedy strategy is utilized to assign combination of mutation strategies and control parameters to each individual of the current population according to the assigned historical experience. The combination of mutation strategies and control parameters corresponding to the maximum value of the assigned historical experience, with (1−ε) probability, is selected, and the population randomly selects a combination with ε probability.

After the population performs once evolution search, the historical experience archive will be updated, in which the oldest record of the historical experience archive is replaced by the new historical experience calculated by

$$E^t = (1 - c) * E^t + c * \text{mean}\left(I^t\right), \tag{9}$$

where c is cumulative degree factor and a constant in interval [0, 1], and mean ($*$) denotes the arithmetic mean operation. The I^t is a vector with the same dimension as E^t, consisted of average nonnegative improvements of fitness value of the individuals that use corresponding combinations for evolution search.

3.3. The Used Mutation Strategies and Control Parameters.

There are three well-known mutation strategies described as formulas (2), (10), and (11), which are also utilized to evolution search in EPSDE. DE/rand/1 (formula (2)) was slower, but more robust than the other strategies that rely on the best-so-far vector. DE/best/2/bin are better than DE/best/1 due to its ability to improve diversity by producing more trial vectors. DE/current-to-rand/1 being a rotation-invariant strategy without crossover can solve rotated problems better than other strategies. Thus, these three mutation strategies are used in ASDE.

DE/best/2 [44]:

$$V_i^t = X_{\text{best}}^t + F * \left(X_{r1}^t - X_{r2}^t\right) + F * \left(X_{r3}^t - X_{r5}^t\right). \tag{10}$$

DE/current-to-rand/1 [45]:

$$U_i^t = X_i^t + K * \left(X_{r1}^t - X_i^t\right) + F * \left(X_{r2}^t - X_{r3}^t\right), \tag{11}$$

where X_{best}^t is the optimal solution of the current population, and K denotes the random chosen from interval [0, 1]. Additionally, in DE/current-to-best/1 strategy, the crossover operator is not performed.

The parameter feasible space of F and CR is equally divided into 5 intervals, i.e., [0, 0.2], [0.2, 0.4], [0.4, 0.6], [0.6, 0.8], and [0.8, 1]. When the specific interval is selected to set a parameter's value, the exact value of the parameter is chosen uniformly randomly from within that interval.

3.4. Algorithm Architecture.

As shown in Figure 3, the algorithm starts with the initialized population and historical experience archive, in which each individual of the initialized population is randomly and uniformly sampled from feasible solution space, and the population features PF^t and the historical experience E of each record of the initialized historical experience archive are initialized as infinity and zeros vector, respectively.

Before the population evolution search, the historical experience E^t with highest population feature similarity is assigned to the current population by calculating the population feature similarity PFS between population features PF^t of current population and population features PF_k of the historical experience archive according to formula (8).

Then, each individual of the population utilizes ε-greedy strategy to select the combination of mutation strategies and control parameters according to the assigned historical experience E^t, and uses the assigned combination to perform mutation, crossover and selection.

After all individuals of the population evolution search, the average nonnegative improvement of fitness value of the individuals is utilized to update the historical experience E^t, and the updated historical experience E^t and the population feature PF^t are used to update the historical experience archive.

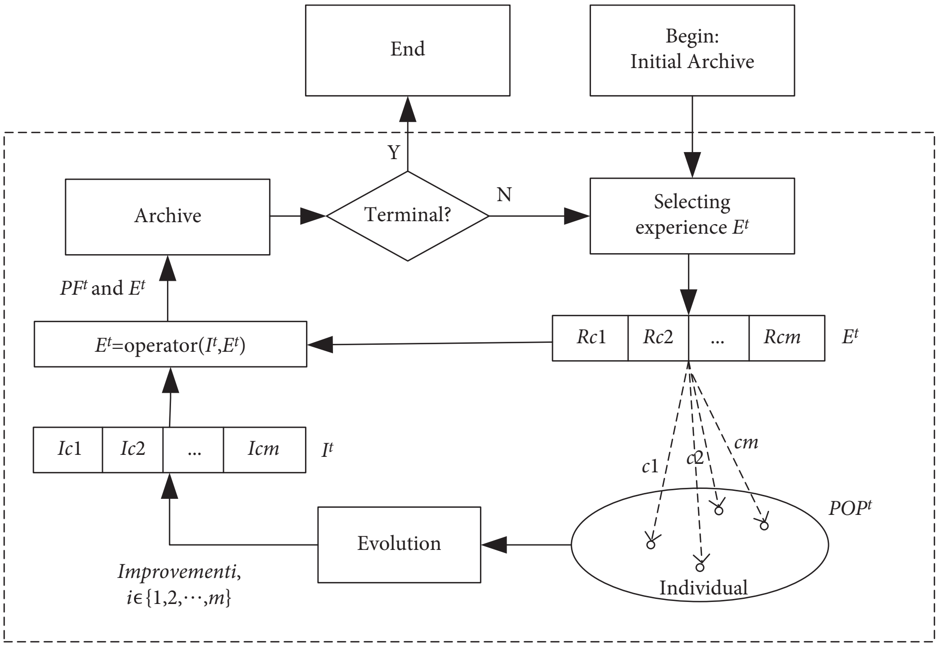


Figure 3: The architecture of the ASDE.

The above process will be repeated until the termination condition is met. The pseudocode of ASDE is shown in Algorithm 1.

4. The Experiment Results of 15 Benchmark Functions of CEC2015

In this section, firstly, ASDE is used to optimize the 15 test functions of CEC2015 [46] that are composed of 2 unimodal functions (F1 and F2), 3 multimodal functions (F3–F5), 3 hybrid functions (F6–F8), and 7 composition functions (F9–F15), compared with some state-of-the-art algorithms (DE [1], EPSDE [31], PSO [47], SLPSO [48], MFO [49], AMFO [50], and RLDE [51]).

Furthermore, 28 benchmark functions of CEC2017 [52] are used to verify the performance of ASDE, which consists of 1 unimodal function F1, 7 simple multimodal functions (F3–F9), 10 hybrid functions (F10–F19), and 10 composition functions (F20–F29). It is noteworthy that F2 of CEC2017 is not used because it is an unstable problem. Some state-of-the-art algorithms are utilized to participate in the comparison, such as, DE [1], EPSDE [31], PSO [47], SLPSO [48], MFO [49], AMFO [50], RLDE [51], jSO [53], LSHADE-SPACMA [54], and EB-LSHADE [55].

The common parameters of all the algorithms are set as follows. The population size is set as 100, and the maximum number of function evaluation is 10000^*D, where D denotes the dimensions of the solved problems. In CEC2015, the dimension D is set as 30. The dimensions D are 10, 30, 50, and 100 for CEC2017. Meanwhile, in ASDE, the size of the historical experience archive is the same to population size, and ε is equal to 0.4. The cumulative degree factor c is set as 0.3 according to the sensitivity analysis of it in Section 4.3. In addition, the other control parameters of the compared algorithm follow their original papers.

In this experiment, the average value and standard deviation of the function error value $(f(g\text{best}) - f(X^*))$ are recorded for verifying the performance of the algorithms, gained by each algorithm with 51 independent runs, where $g$best is the best solution obtained by the algorithm in a run and X^* denotes the theoretical global optimum of the benchmark functions. The maximum number of function evaluations (FES) is set to 10000^*D for both CEC2013 and CEC2017. Wilcoxon's rank sum test at a 5% significance level was utilized to generate statistically reliable results for CEC2015, and the Friedman test is used in CEC2017. According to Wilcoxon signed-rank test, R^+ is the sum of ranks for the functions in which ASDE outperforms the compared algorithm and R^- is the sum of ranks for the opposite. Larger ranks demonstrate larger performance discrepancy.

4.1. Comparisons on Solution Accuracy. In each row of Table 1, the mean values over 25 independent runs are shown in the first line, and the standard deviations are presented in the second line. The P value and H value of the nonparametric statistical test (Wilcoxon's rank sum test) with a significance level $\alpha = 0.05$ are given in the third and fourth lines. The symbol '‡' is tagged in the back of the mean value produced by the algorithm that is significantly worse than ASDE. If ASDE is worse than other algorithms, a 'ξ' is added in the back of the mean value of corresponding algorithm. The symbol '~' demonstrates that there is no significant difference between ASDE and compared algorithm. At the last row of the table, a summary of total number of "‡," "ξ"," and "~" is presented. Additionally, the best solution is highlighted in bold.

It is clearly seen from Table 1 that ASDE gains the better performance than EPSDE on 7 functions composed of 1

```
Begin
  All $\text{std}_f$, $\text{std}_{\text{pop}}$ and stag of the historical experience archive are set infinity, and all historical
  experience ($E$) of the historical experience archive are equal to 0;
  pos = 1;
  Initialize uniformly random population pop and calculate fitness value of population;
  Set $t = 0$;
  While termination condition is not met
    $t = t + 1$;
    Calculate population feature $PF^t$ by using formulas (5)–(7);
    Calculate population feature similarity $PFS$ by using formula (8);
    $[\sim, I] = \min\ (PFS)$;
    $E^t = E_I$, where $E_I$ is $I$th record of the historical experience archive.
    for $i = 1$ to $NP$
      $a_i$ is selected with $\varepsilon$-greedy strategy according to $E^t$, $a_i \in \{1, 2, \ldots, 75\}$;
      $MS_i = \text{floor}((a_i - 1)/25 + 1)$, where $MS_i$ denotes the mutation strategy used by $i$ th individual;
      $F_i = \text{floor}\ (\text{mod}\ (a_i - 1, 25)/5)\ {}^*\ 0.2 + \text{rand}\ (0, 1)\ {}^*\ 0.2$;
      $CR_i = \text{mod}\ (a_i - 1, 2)\ {}^*\ 0.2 + \text{rand}\ (0, 1)\ {}^*\ 0.2$;
      Mutation, crossover and generation of trial vector $u$ using $MS_i$, $F_i$ and $CR_i$;
    end for
    for $i = 1$ to $NP$
      if $f(u_i) < f(x_i)$
        $x_i = u_i$;
        $I^t\ (a_i, :) = [R^t\ (a_i), f(x_i) - f(u_i)]$;
      end if
    end for
    Updating the $E^t$ by using formula (9);
    Updating the historical experience archive by using $E^t$ and $PF^t$;
    pos = pos + 1;
    if pos > $H$
      pos = 1;
    end if
  end while
End
```

Algorithm 1: Pseudocode of ASDE.

multimodal function (F5), 3 hybrid functions (F6–F8), and 3 composition functions (F11–F13), which indicates that the proposed historical experience archive with population features and the updating and utilization mechanism of the historical experience can provide the better combination of mutation strategies and control parameters than EPSDE. ASDE is worse than EPSDE on 1 functions F14, which may be because the ε-greedy strategy ($\varepsilon = 0.4$) is slightly poor in the exploitation ability of the population. In addition, ASDE is similar to the EPSDE on 7 functions (F1–F4, F9, F10, and F15).

ASDE obtains the best performance on 5 functions (F3, F7, F11, F13, and F15), which shows that the ASDE is promising in solving complex problems. EPSDE is the best on 3 functions (F1–F3), and DE achieves the best performance on 5 functions (F6, F8, F10, F14, and F15). Meanwhile, PSO only is the best on function F3, and SLPSO outperforms other compared algorithms on 5 functions (F4, F5, F9, F12, and F15). Additionally, MFO gains the best performance on function (F15).

Table 2 shows the average ranks according to Friedman test for the compared algorithms using CEC2017. The best ranks are shown in bold, and the second ranks are underlined. From Table 2, we can see that P value generated by Friedman test for all dimensions are less than 0.05. Therefore, it can be concluded that there is a significant difference between the performances of the algorithms.

It can be clearly seen from Table 2 that ASDE is ranked first for 10 dimensions and ranked second for 30 and 100 dimensions. Regarding mean ranking, LSHADE-SPACMA gains the best ranking, and ASDE obtained the second ranking.

According to Wilcoxon's test shown in Table 3, ASDE is significantly better than AMFO, DE, EPSDE, MFO, PSO, and RLDE for all dimensions. On the other hand, there is no significant difference between ASDE and LSHADE-SPACMA for all dimensions. The performance of ASDE is similar to that of jSO for $D = 30$, 50, and 100. There is no significant difference between ASDE and both of EBL-SHADE and SLPSO for $D = 50$ and 100.

4.2. *The Comparison Results of Convergence Speed.*

In Figure 4, the vertical axis is the nature logarithm of the mean value over independent 25 runs, and the horizontal axis is the sampling point where 30 sampling points are taken from FES = 1000 and mod (FES, 10000) = 0.

Figure 4 clearly shows that ASDE obtains the best performance on 5 functions (F7, F9, F11, F13, and F15), which demonstrates that ASDE is effective to optimize some complex

Table 1: Results of solution accuracy gained by eight algorithms on 15 benchmark functions of CEC2015.

Func.	Results	ASDE	DE	EPSDE	PSO	SLPSO	MFO	AMFO	RLDE
F1	Mean	3.78E+04	6.03E+05‡	**3.36E+04**~	2.84E+07‡	3.87E+05‡	1.28E+06‡	2.84E+08‡	4.80E+07‡
	Std	2.50E+04	5.01E+05	**4.06E+04**	5.01E+07	2.53E+05	6.84E+05	5.97E+07	4.26E+07
	P value	—	9.36E−06	6.78E−01	9.19E−03	3.06E−07	8.46E−08	7.79E−22	8.35E−06
	H value	—	1	0	1	1	1	1	1
F2	Mean	3.39E−10	8.62E+03‡	**3.87E−14**~	4.07E+09‡	4.70E+03‡	6.62E+03‡	2.11E+10‡	2.71E+09‡
	Std	8.57E−10	2.06E+04	**1.81E−14**	3.16E+09	5.01E+03	5.82E+03	2.95E+09	6.21E+09
	P value	—	4.68E−02	5.98E−02	1.17E−06	9.24E−05	3.05E−05	6.96E−23	3.91E−02
	H value	—	1	0	1	1	1	1	1
F3	Mean	**2.06E+01**	2.08E+01‡	**2.06E+01**~	**2.06E+01**~	2.09E+01‡	2.00E+01‡	2.10E+01‡	2.10E+01‡
	Std	**1.91E−01**	1.43E−01	**3.33E−02**	**2.77E−01**	5.51E−02	1.86E−02	5.60E−02	9.63E−02
	P value	—	3.64E−03	7.99E−01	6.29E−01	1.25E−09	4.30E−14	4.10E−09	3.45E−10
	H value	—	1	0	0	1	1	1	1
F4	Mean	1.01E+02	5.93E+01ξ	1.16E+02~	6.63E+01	**1.76E+01** ξ	1.02E+02~	3.00E+02‡	2.10E+02‡
	Std	3.89E+01	2.02E+01	9.65E+00	1.73E+01	**5.22E+00**	2.52E+01	1.74E+01	6.04E+01
	P value	—	7.44E−05	1.06E−01	2.05E−04	4.15E−10	6.49E−01	2.42E−16	8.91E−09
	H value	—	1	0	1	1	0	1	1
F5	Mean	3.14E+03	5.85E+03‡	5.34E+03‡	3.04E+03~	**8.16E+02** ξ	3.15E+03~	7.09E+03‡	5.23E+03‡
	Std	6.09E+02	8.57E+02	2.61E+02	3.74E+02	**4.83E+02**	6.28E+02	3.17E+02	1.22E+03
	P value	—	1.48E−12	9.34E−14	4.99E−01	6.71E−14	1.33E−01	5.48E−21	9.06E−08
	H value	—	1	1	0	1	0	1	1
F6	Mean	4.50E+03	**1.48E+03** ξ	1.04E+04‡	3.13E+05‡	7.07E+04‡	1.43E+05‡	9.27E+06‡	1.55E+06‡
	Std	3.04E+03	**5.56E+02**	1.29E+04	6.37E+05	5.75E+04	8.20E+04	2.89E+06	1.98E+06
	P value	—	2.25E−05	3.87E−02	2.35E−02	6.72E−06	4.16E−08	2.25E−13	6.61E−04
	H value	—	1	1	1	1	1	1	1
F7	Mean	**3.76E+00**	6.41E+00‡	1.44E+01‡	2.50E+01‡	1.09E+01‡	8.41E+00‡	5.36E+01‡	2.61E+01‡
	Std	**1.26E+00**	1.31E+00	9.51E−01	1.78E+01	8.06E−01	1.30E+00	7.26E+00	1.97E+01
	P value	—	1.32E−06	8.83E−22	4.08E−06	2.09E−18	7.95E−15	2.31E−24	7.81E−06
	H value	—	1	1	1	1	1	1	1
F8	Mean	3.90E+02	**2.96E+02**~	1.21E+03‡	7.35E+04‡	3.69E+04‡	8.32E+04‡	2.07E+06‡	2.24E+05‡
	Std	2.49E+02	**1.99E+02**	3.87E+02	9.73E+04	2.35E+04	6.20E+04	8.12E+05	1.88E+05
	P value	—	2.08E−01	1.09E−08	9.73E−04	5.24E−08	4.78E−06	3.55E−12	4.01E−06
	H value	—	0	1	1	1	1	1	1
F9	Mean	1.03E+02	1.04E+02‡	1.03E+02~	1.44E+02‡	**1.02E+02** ξ	1.05E+02‡	1.79E+02‡	1.23E+02‡
	Std	1.76E−01	4.66E−01	1.51E−01	4.56E+01	**2.95E−01**	5.46E−01	9.92E+00	4.49E+01
	P value	—	2.32E−09	8.56E−10	1.51E−04	7.23E−04	1.07E−17	2.88E−27	3.08E−02
	H value	—	1	1	1	1	1	1	1
F10	Mean	1.35E+03	**5.00E+02** ξ	1.59E+03~	6.37E+05‡	8.45E+04‡	1.16E+05‡	5.63E+06‡	1.56E+06‡
	Std	9.09E+02	**2.28E+02**	9.93E+02	1.38E+06	4.70E+04	8.82E+04	2.24E+06	2.77E+06
	P value	—	6.04E−05	3.54E−01	3.00E−02	4.44E−09	3.13E−05	8.55E−12	9.37E−03
	H value	—	1	0	1	1	1	1	1
F11	Mean	**4.91E+02**	1.25E+03‡	1.14E+03‡	6.53E+02‡	5.12E+02~	6.93E+02‡	8.41E+02‡	1.01E+03‡
	Std	**1.31E+02**	1.17E+02	5.63E+01	3.01E+02	4.41E+01	2.07E+02	1.94E+02	2.75E+02
	P value	—	1.78E−17	5.33E−17	2.27E−02	4.54E−01	4.29E−09	1.00E−02	1.11E−09
	H value	—	1	1	1	0	1	1	1
F12	Mean	1.06E+02	1.09E+02‡	1.07E+02‡	1.32E+02‡	**1.05E+02** ξ	1.11E+02‡	1.47E+02‡	1.15E+02‡
	Std	6.34E−01	1.59E+00	3.41E−01	1.14E+01	**6.37E−01**	1.99E+00	3.59E+00	9.14E+00
	P value	—	1.99E−08	3.02E−08	3.43E−11	9.07E−05	2.79E−10	2.68E−24	2.37E−05
	H value	—	1	1	1	1	1	1	1
F13	Mean	**1.09E+02**	1.10E+02~	1.31E+02‡	1.22E+02‡	1.20E+02‡	1.22E+02‡	1.45E+02‡	1.37E+02‡
	Std	**6.18E+00**	1.15E+01	1.84E+00	6.54E+00	3.69E+00	6.77E+00	2.41E+00	5.15E+00
	P value	—	8.26E−01	2.43E−14	3.07E−07	5.24E−08	1.75E−06	2.97E−16	2.18E−14
	H value	—	0	1	1	1	1	1	1
F14	Mean	3.23E+04	**6.59E+03** ξ	1.28E+04ξ	4.14E+04‡	3.33E+04‡	3.35E+04‡	5.15E+04‡	3.74E+04‡
	Std	9.94E+02	**1.01E+04**	1.20E+04	4.14E+03	1.00E+03	1.09E+03	2.50E+03	3.77E+03
	P value	—	1.49E−12	1.88E−08	5.47E−11	4.56E−04	3.96E−06	2.99E−23	1.73E−06
	H value	—	1	1	1	1	1	1	1
F15	Mean	**1.00E+02**	**1.00E+02**~	**1.00E+02**~	1.25E+02‡	**1.00E+02**~	**1.00E+02**~	1.26E+03‡	7.35E+02‡
	Std	**4.35E−14**	**5.17E−04**	**4.35E−14**	9.97E+00	**1.10E−13**	**1.66E−13**	6.43E+02	1.58E+03
	P value	—	2.57E−02	NAN	5.89E−12	5.44E−03	2.82E−05	1.13E−12	4.56E−02
	H value	—	1	NAN	1	1	1	1	1
‡/ξ/~		—	8/4/3	7/1/7	13/0/2	9/4/2	12/0/3	15/0/0	15/0/0

Table 2: Average ranks for all algorithms across 28 problems and all dimensions using CEC2017.

Algorithms	10D	30D	50D	100D	Mean ranking	Rank
ASDE	**2.68**	3.46	4.32	4.07	3.63	2
AMFO	9.04	10.61	10.75	10.93	10.33	10
DE	3.96	5.05	5.64	6.04	5.17	6
EBLSHADE	4.32	4.66	4.64	4.93	4.64	4
EPSDE	5.41	5.44	5.78	6.16	5.70	7
jSO	5.35	4.25	3.95	4.18	4.43	3
LSHADE-SPACMA	3.39	**2.02**	**1.52**	**1.63**	**2.14**	**1**
MFO	7.13	7.75	7.25	6.82	7.24	8
PSO	7.39	7.71	7.78	7.71	7.65	9
RLDE	10.71	10.21	9.75	9.32	10.00	11
SLPSO	6.61	4.82	4.1	4.21	4.94	5

Table 3: Wilcoxon's test between ASDE and other algorithms for $D = 10$, 30, 50, and 100 using CEC2017.

D	Algorithms	R^+	R^-	P value
10	AMFO	406.0	0.0	**0.0000**
	DE	310.5	67.5	0.0034
	EBLSHADE	320.5	85.5	0.0070
	EPSDE	359.5	18.5	**0.0000**
	jSO	315.5	62.5	0.0023
	LSHADE-SPACMA	256.0	150.0	0.2178
	MFO	406.0	0.0	**0.0000**
	PSO	402.0	4.0	**0.0000**
	RLDE	406.0	0.0	**0.0000**
	SLPSO	362.0	44.0	0.0003
30	AMFO	406.0	0.0	**0.0000**
	DE	321.0	57.0	0.0014
	EBLSHADE	260.0	118.0	0.0258
	EPSDE	364.0	42.0	0.0002
	jSO	255.0	123.0	0.1101
	LSHADE-PACMA	72.5	333.5	1.0000
	MFO	406.0	0.0	**0.0000**
	PSO	406.0	0.0	**0.0000**
	RLDE	406.0	0.0	**0.0000**
	SLPSO	301.0	105.0	0.0248
50	AMFO	406.0	0.0	**0.0000**
	DE	333.0	73.0	0.00030
	EBLSHADE	226.0	180.0	0.5925
	EPSDE	366.0	40.0	0.0002
	jSO	159.0	247.0	1.0000
	LSHADE-SPACMA	8.0	398.0	1.0000
	MFO	406.0	0.0	**0.0000**
	PSO	401.0	5.0	**0.0000**
	RLDE	406.0	0.0	**0.0000**
	SLPSO	247.0	159.0	0.3109
100	AMFO	406.0	0.0	**0.0000**
	DE	359.0	47.0	0.0004
	EBLSHADE	244.0	162.0	0.3447
	EPSDE	368.0	38.0	0.0001
	jSO	156.0	250.0	1.0000
	LSHADE-SPACMA	7.0	399.0	1.0000
	MFO	406.0	0.0	**0.0000**
	PSO	400.0	4.0	**0.0000**
	RLDE	406.0	0.0	**0.0000**
	SLPSO	235.0	171.0	0.4593

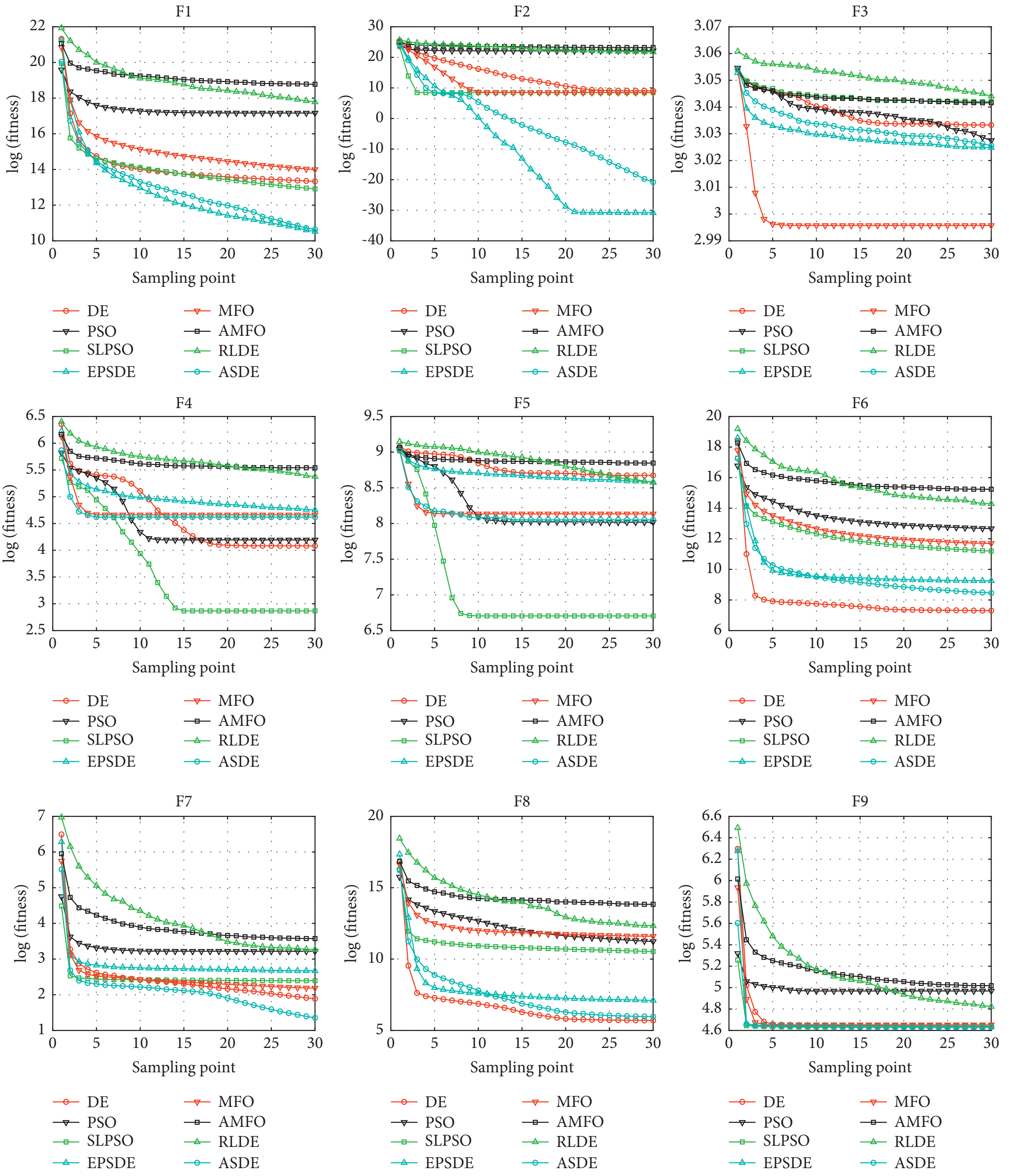
problems. Although ASDE gains poor convergence performance on functions F1, it has strong global exploration ability and achieves the best convergence accuracy. This may be because the proposed mechanism can assign autonomously the combination of mutation strategies and control parameters to the population so that the population can use the more exploratory mutation strategies and control parameters to search.

Additionally, DE is best on 5 functions (F6, F8, F10, F14, and F15), and EPSDE gains the best convergence speed on 3 functions (F1, F2, and F15). MFO obtains the best convergence speed on 2 functions (F3 and F15), and SLPSO is the best convergence speed on 5 functions (F4, F5, F9, F12, and F15). PSO and RLDE have poor convergence speed on all benchmark functions.

4.3. *Sensitivity Analysis of the Parameter* **c**.

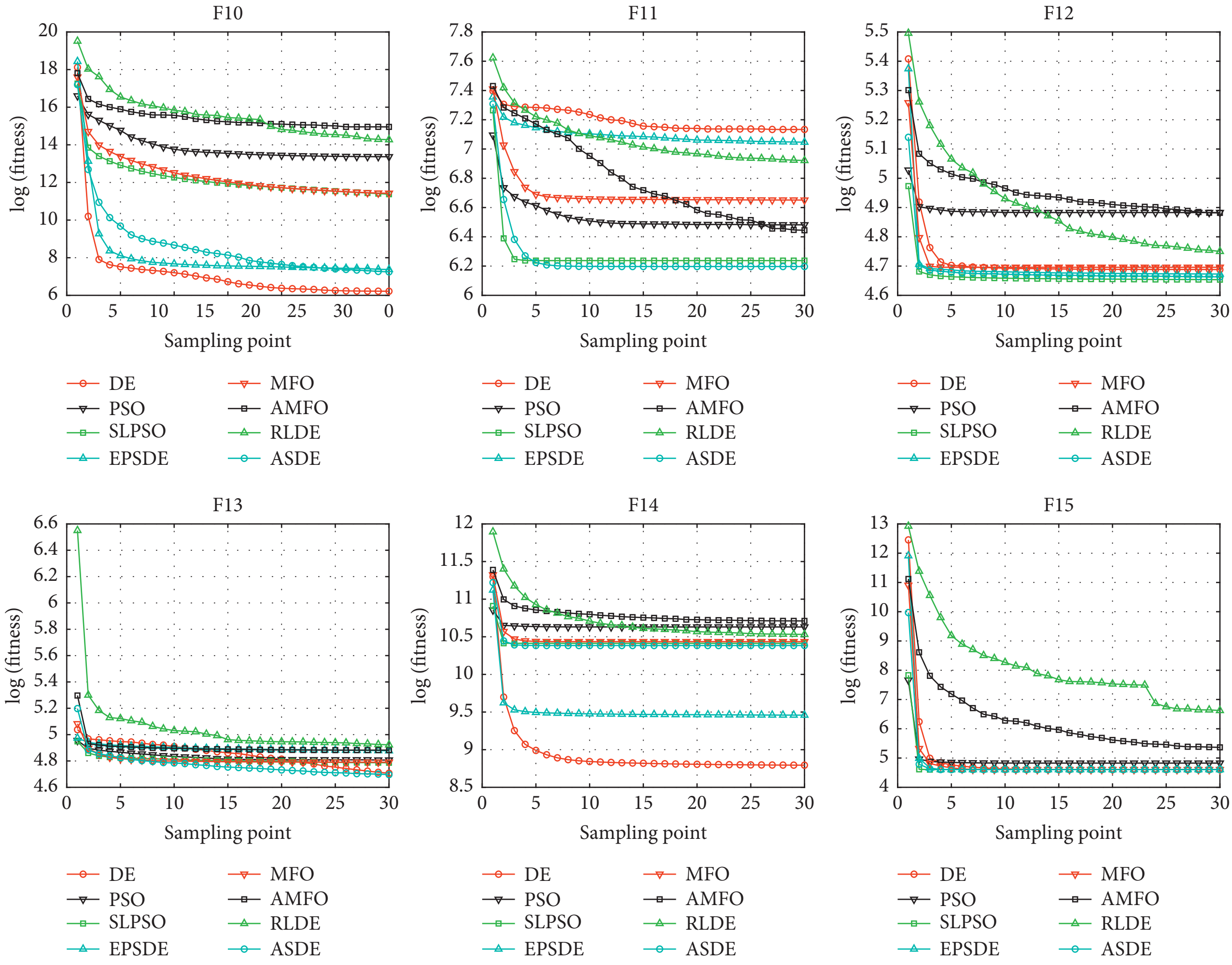
The parameter c determines the impact level of the historical experience on the current population; that is, in the historical experience, the larger the value of the parameter c, the greater the proportion of the reward of each combination used in the current population. To find out a good choice of the parameter c, ASDE with different parameter $c = \{0.1, 0.2, \ldots, 1\}$ is utilized to optimize 15 benchmark problems of CEC2015. The other parameter settings of the algorithm are the same as the settings described above. Table 4 summarizes the results of average errors over 25 independent runs. For each problem, the best result of ASDE with different values of the parameter c is displayed in bold font. At the last row of the table, an average ranking of ASDE with different values of the parameter c is presented.

Table 4 clearly shows that the c value affects the performance of algorithms, and the same c value gains different performance on different functions. ASDE with $c = 0.3$ obtains the best performance on 6 functions (F3, F8, F9, F12, F14, and F15), and its average ranking is the best. In addition, although ASDE with $c = 1$ obtains the best performance on 9 functions (F1–F4, F6, F9, F12, F13, and F15), it makes the historical experience more volatile, because evolutionary algorithm search has probability features. Thus, to make the algorithm more robust and make the historical



(a)

Figure 4: Continued.



(b)

Figure 4: Convergence performance of the eight compared algorithms on 25 functions of CEC2005.

experience more stable, $c = 0.3$ is recommended for the ASDE algorithm.

4.4. Comparison Results of Time Complexity. In this section, the time complexity of ASDE is evaluated and is expressed as follows:

(1) The complexity of population initialization is $O(N)$.

(2) Calculating the fitness value costs $O(N)$.

(3) Calculating the diversity of population costs $O(N^2)$.

(4) The feature similar between current population and the records of the historical experience archive needs $O(N)$.

(5) Assigning control parameters and mutation strategies for each individual costs $O(N)$.

(6) Generating the individual to be updated cost $O(N)$.

(7) Calculating the fitness value of offspring population costs $O(N)$.

(8) The update of the historical experience archive is $O(N)$.

The overall time complexity of ASDE is shown as follows:

$$\begin{aligned} &O(N) + O(N) + O\left(N^2\right) + O(N) + O(N) + O(N) \\ &\quad + O(N) + O(N) = 7O(N) + O\left(N^2\right). \end{aligned} \tag{12}$$

The total comparisons of average time complexity of 28 functions for $D = 100$ using CEC2017 in one iteration about 11 algorithms are displayed in Figure 5 in the form of bar plot. Figure 5 clearly shows that the mean CPU time of ASDE is slightly worse than DE, jSO, PSO, and SLPSO, and slightly better than MFO and RLDE. In addition, the mean CPU time of ASDE is significantly better than AMFO, EBLSHADE, EPSDE, and LSHADE-SPACMA.

5. The Application about Parameter Extraction of Photovoltaic Models

As we all know, solar energy is one of the most important renewable energy sources [56], and its main practical application is photovoltaic (PV) power generation [57], because solar energy can be directly converted into electricity through the PV system. However, the conversion efficiency of PV model is

Table 4: Results of solution accuracy obtained by ASDE with different **c** value on 15 benchmark functions of CEC2015.

Fs	Result	c = 0.1	c = 0.2	c = 0.3	c = 0.4	c = 0.5	c = 0.6	c = 0.7	c = 0.8	c = 0.9	c = 1
F1	Mean	4.32E+04	5.12E+04	3.78E+04	4.27E+04	4.34E+04	3.90E+04	2.04E+04	1.95E+04	1.53E+04	**7.93E+03**
	Std	2.72E+04	5.31E+04	2.50E+04	3.62E+04	2.82E+04	2.85E+04	1.54E+04	1.61E+04	1.22E+04	**5.77E+03**
F2	Mean	5.79E–E–09	7.26E–10	3.39E–10	1.26E–10	1.36E–09	3.64E–10	3.06E–10	9.80E–12	2.30E–11	**4.02E–12**
	Std	2.65E–08	1.96E–09	8.57E–10	2.15E–10	5.40E–09	1.28E–09	1.50E–09	3.78E–11	1.01E–10	**1.80E–11**
F3	Mean	**2.06E+01**	**2.06E+01**	**2.06E+01**	**2.06E+01**	2.07E+01	2.07E+01	2.07E+01	**2.06E+01**	2.07E+01	**2.06E+01**
	Std	**1.51E–01**	**2.52E–01**	**1.91E–01**	**2.66E–01**	1.51E–01	1.12E–01	5.95E–02	**8.38E–02**	5.16E–02	**1.47E–01**
F4	Mean	1.38E+02	1.29E+02	1.01E+02	7.96E+01	7.59E+01	7.65E+01	8.06E+01	6.83E+01	7.50E+01	**6.46E+01**
	Std	6.07E+01	3.97E+01	3.89E+01	3.58E+01	2.84E+01	2.85E+01	2.47E+01	1.79E+01	2.28E+01	**2.56E+01**
F5	Mean	3.08E+03	**2.90E+03**	3.14E+03	3.32E+03	3.18E+03	2.95E+03	2.94E+03	2.99E+03	2.94E+03	2.83E+03
	Std	6.78E+02	**7.32E+02**	6.09E+02	6.07E+02	6.79E+02	5.54E+02	4.93E+02	6.92E+02	4.84E+02	7.09E+02
F6	Mean	4.48E+03	4.54E+03	4.50E+03	5.35E+03	4.62E+03	5.59E+03	5.21E+03	4.84E+03	4.35E+03	**3.30E+03**
	Std	2.27E+03	2.99E+03	3.04E+03	7.82E+03	3.36E+03	5.25E+03	4.53E+03	4.55E+03	3.11E+03	**2.42E+03**
F7	Mean	4.50E+00	3.70E+00	3.76E+00	4.01E+00	3.65E+00	**3.46E+00**	3.81E+00	5.42E+00	4.32E+00	4.79E+00
	Std	1.88E+00	1.46E+00	1.26E+00	1.68E+00	1.05E+00	**8.40E–01**	9.85E–01	4.54E+00	1.06E+00	1.23E+00
F8	Mean	5.82E+02	4.63E+02	**3.90E+02**	5.69E+02	1.10E+03	6.72E+02	5.11E+02	4.61E+02	5.48E+02	5.17E+02
	Std	5.46E+02	4.46E+02	**2.49E+02**	5.22E+02	3.14E+03	1.19E+03	3.21E+02	2.45E+02	4.35E+02	4.54E+02
F9	Mean	**1.03E+02**	1.12E+02	**1.03E+02**	**1.03E+02**	1.09E+02	1.09E+02	**1.03E+02**	1.08E+02	1.03E+02	**1.03E+02**
	Std	**2.98E–01**	4.77E+01	**1.76E–01**	**1.70E–01**	2.91E+01	3.11E+01	**1.65E–01**	2.84E+01	2.29E–01	**1.58E–01**
F10	Mean	1.30E+03	1.50E+03	1.35E+03	1.25E+03	1.30E+03	**1.12E+03**	1.34E+03	1.17E+03	1.13E+03	1.14E+03
	Std	4.04E+02	6.56E+02	9.09E+02	5.11E+02	5.26E+02	**3.40E+02**	5.17E+02	4.20E+02	4.15E+02	3.06E+02
F11	Mean	5.21E+02	**4.79E+02**	4.91E+02	5.35E+02	5.38E+02	5.46E+02	6.46E+02	6.01E+02	6.62E+02	6.93E+02
	Std	2.28E+02	**1.52E+02**	1.31E+02	1.31E+02	1.53E+02	1.64E+02	1.73E+02	2.05E+02	1.99E+02	1.90E+02
F12	Mean	1.07E+02	**1.06E+02**	**1.06E+02**	**1.06E+02**	**1.06E+02**	**1.06E+02**	**1.06E+02**	**1.06E+02**	1.06E+02	**1.06E+02**
	Std	2.11E+00	**8.33E–01**	**6.34E–01**	**6.31E–01**	**5.73E–01**	**7.26E–01**	**9.50E–01**	**7.30E–01**	9.58E–01	**9.15E–01**
F13	Mean	1.08E+02	1.12E+02	1.09E+02	1.09E+02	**1.07E+02**	1.10E+02	1.09E+02	**1.07E+02**	1.04E+02	**1.07E+02**
	Std	5.12E+00	8.41E+00	6.18E+00	3.85E+00	**5.24E+00**	3.99E+00	5.43E+00	**6.75E+00**	7.46E+00	**5.65E+00**
F14	Mean	3.29E+04	3.30E+04	**3.23E+04**	3.25E+04	3.28E+04	3.25E+04	3.30E+04	3.28E+04	3.38E+04	3.34E+04
	Std	1.54E+03	8.93E+02	**9.94E+02**	1.09E+03	1.00E+03	8.77E+02	1.38E+03	1.09E+03	1.62E+03	1.49E+03
F15	Mean	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	1.00E+02	**1.00E+02**
	Std	**4.35E–14**	**4.35E–14**	**4.35E–14**	**5.69E–14**	**5.69E–14**	**4.35E–14**	**5.69E–14**	**6.77E–14**	4.35E–14	**5.69E–14**
Ran.	—	6.40	5.67	4.53	5.26	6.20	5.87	6.13	5.13	5.07	4.73

greatly affected by model parameters. To deal with this problem, the ASDE is used to extract the parameters of PV models.

5.1. PV Models and Fitness Functions.

The single diode and double diode models [58] are the most commonly used PV models, which can explain the current-voltage features of the PV systems. The single diode model, the diode model, the PV module and the fitness function are modeled as follows.

5.1.1. Single-Diode Model.

The output current and voltage of the single diode model is calculated by using [59]

$$I = I_{ph} - I_d\left[\exp\left(\frac{V + IR_s}{aV_t}\right) - 1\right] - \frac{V + IR_s}{R_{sh}}, \tag{13}$$

where R_s and R_{sh} are the series resistance and the shunt resistance, respectively, and a stands for the diode ideal factor. I and V stands for the output current and the output voltage of the single diode model, respectively. I_{ph} and I_d are the photogenerated current and the diode current, respectively. V_t represents the junction thermal voltage, generated by using

$$V_t = \frac{k \cdot T}{q}, \tag{14}$$

where k denotes the Boltzmann constant (1.3806503×10^{-23} J/K), and T stands for the temperature of junction in Kelvin. q represents the electron charge ($1.60217646\times 10^{-19}$C).

The unknown parameters (R_s, R_{sh}, a, I_{ph}, and I_d) of single diode model will be extracted.

5.1.2. Double-Diode Model.

The relationship of output current and voltage of double diode model is described as

$$\begin{aligned} I = I_{ph} &- I_{d1}\left[\exp\left(\frac{V + IR_s}{a_1 V_t}\right) - 1\right] \\ &- I_{d2}\left[\exp\left(\frac{V + IR_s}{a_2 V_t}\right) - 1\right] - \frac{V + IR_s}{R_{sh}}, \end{aligned} \tag{15}$$

where $I_{di}, i \in \{1,2\}$ denotes the current of the i-th diode, and $a_i, i \in \{1,2\}$ represents the i-th diode ideal factor.

There are five unknown parameters (R_s, R_{sh}, a_1, a_2, I_{ph}, I_{d1}, and I_{d2}) of the double-diode model that is extracted.

5.1.3. PV Module.

The output current and voltage of the PV module is described as [60]

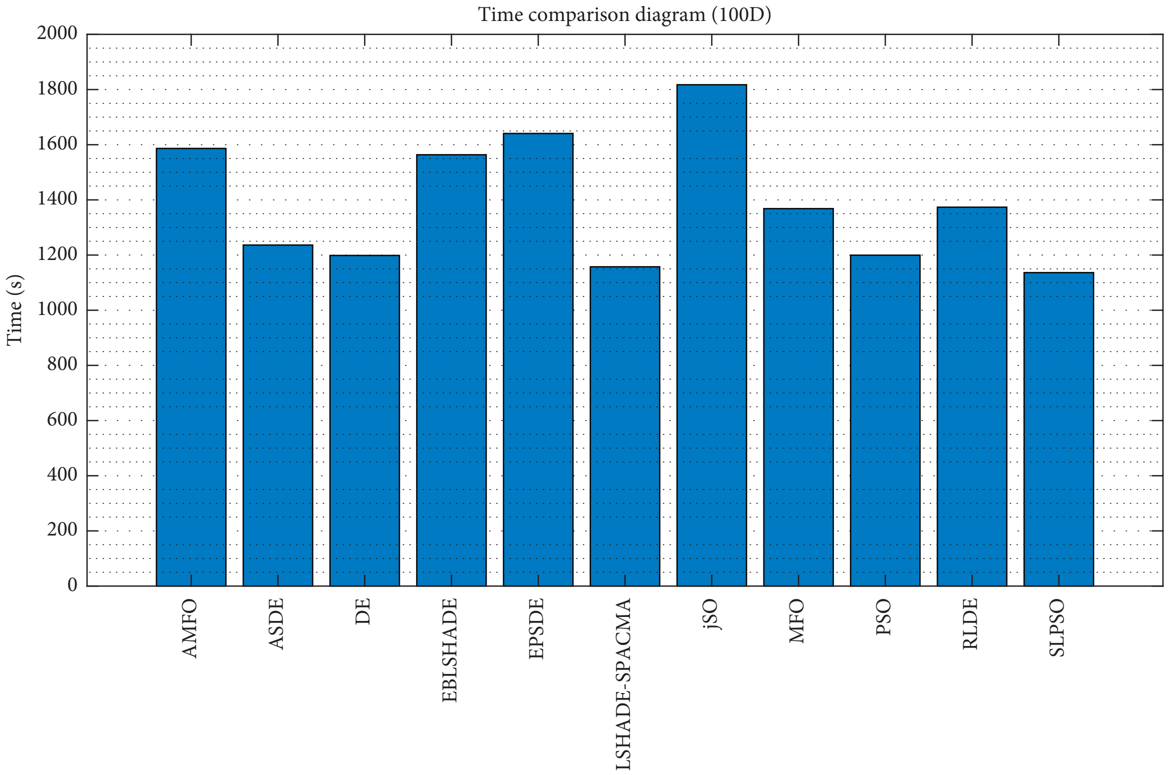


Figure 5: Mean CPU time of 11 algorithms.

$$I = I_{ph}N_p - I_dN_p\left[\exp\left(\frac{V + IR_sN_s/N_p}{aN_sV_t}\right) - 1\right] - \frac{V + IR_sN_s/N_p}{R_{sh}N_s/N_p}, \quad (16)$$

where N_s is the number of solar cells that are connected in series, and N_p denotes the number of solar cells that are connected in parallel. In the experiment, N_p is 1, because the PV modules are all in series. Thus, the output current and voltage of the used PV module is addressed as

$$I = I_{ph} - I_dN_p\left[\exp\left(\frac{V + IR_sN_s}{aN_sV_t}\right) - 1\right] - \frac{V + IR_sN_s}{R_{sh}N_s}. \quad (17)$$

The parameters (R_s, R_{sh}, a, I_{ph}, and I_d) of PV module are needed to extracted.

5.1.4. Fitness Functions. Minimizing the error between simulated and measured current data is the target of parameter extraction of PV models. The absolute error current (AEC) of the individuals is calculated as follows [61].

The AEC of the single diode model is calculated by

$$AEC = \left|I_{ph} - I_d\left[\exp\left(\frac{V + IR_s}{aV_t}\right) - 1\right] - \frac{V + IR_s}{R_{sh}} - I\right|. \quad (18)$$

The AEC of the double diode model is generated by

$$AEC = \left|I_{ph} - I_{d1}\left[\exp\left(\frac{V + IR_s}{a_1V_t}\right) - 1\right] - I_{d2}\left[\exp\left(\frac{V + IR_s}{a_2V_t}\right) - 1\right] - \frac{V + IR_s}{R_{sh}} - I\right|. \quad (19)$$

The AEC of the PV module is created as

$$\text{AEC} = \left|I_{ph} - I_dN_p\left[\exp\left(\frac{V + IR_sN_s}{aN_sV_t}\right) - 1\right] - \frac{V + IR_sN_s}{R_{sh}N_s} - I\right|. \quad (20)$$

To quantify the overall error between the simulated and measured current, the root mean square error is used as the fitness function, addressed as

$$f(x) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\text{AEC}^2}, \quad (21)$$

where x represents the feasible solution consisted of the unknown parameters, and N is the number of measured current data. It is clearly shown that the smaller the fitness value, the more accurate the extracted parameters.

5.2. Parameters Setting. The current-voltage data of single (PV–F1) and double (PV–F2) diode models are obtained from [62], measured on a 57 mm diameter commercial

Table 5: The feasible range of parameters to be extracted.

Parameters	PV-F1/F2		PV-F3		PV-F4		PV-F5	
	LB	UB	LB	UB	LB	UB	LB	UB
I_{ph}	0	1	0	2	0	2	0	8
I_d, I_{d1}, I_{d2}	0	1	0	50	0	50	0	50
R_s	0	0.5	0	2	0	0.36	0	0.36
R_{sh}	0	100	0	2000	0	1000	0	1500
a, a_1, a_2	1	2	1	50	1	60	1	50

Table 6: Solution accuracy of ASDE and compared algorithms on PV problems.

Func.	Results	ASDE	DE	EPSDE	PSO	SLPSO	MFO	AMFO	RLDE
PV-F1	Mean	**9.86E−04**	1.76E−03ǂ	1.31E−03ǂ	2.25E−01ǂ	1.78E−03ǂ	1.88E−03ǂ	9.66E−03ǂ	1.21E−02ǂ
	Std	**3.04E−13**	4.10E−04	4.32E−04	8.50E−17	3.57E−04	3.83E−04	3.68E−03	9.17E−03
	P value	—	1.47E−09	9.23E−04	1.47E−286	6.44E−11	2.15E−11	1.81E−11	2.80E−06
	H value	—	1	1	1	1	1	1	1
PV-F2	Mean	**9.85E−04**	1.45E−03ǂ	1.91E−03ǂ	2.52E−01ǂ	2.55E−03ǂ	1.85E−03ǂ	8.49E−03ǂ	8.60E−03ǂ
	Std	**1.39E−06**	4.36E−04	3.71E−04	1.37E−01	4.56E−04	4.79E−04	2.67E−03	5.82E−03
	P value	—	1.93E−05	5.57E−12	2.44E−09	6.00E−15	3.85E−09	4.58E−13	9.05E−07
	H value	—	1	1	1	1	1	1	1
PV-F3	Mean	2.70E−03	6.89E−03ǂ	3.87E−03ǂ	4.97E + 00ǂ	2.58E−03~	**2.49E−03**~	1.12E−02ǂ	2.80E−02ǂ
	Std	1.35E−03	1.28E−03	7.78E−04	2.72E−15	2.30E−05	**8.36E−05**	3.75E−03	1.78E−02
	P value	—	5.37E−12	2.05E−03	2.67E−87	6.84E−01	4.56E−01	2.79E−10	2.84E−07
	H value	—	1	1	1	0	0	1	1
PV-F4	Mean	1.74E−03	6.30E−02ǂ	**8.80E−06** ξ	3.11E−01ǂ	1.06E−02ǂ	6.87E−03ǂ	1.64E−02ǂ	2.19E−01ǂ
	Std	1.33E−05	2.75E−02	**1.54E−05**	1.22E−16	4.62E−03	2.48E−03	5.71E−03	1.05E−01
	P value	—	5.74E−11	5.40E−48	1.52E−106	9.97E−10	2.39E−10	2.91E−12	2.65E−10
	H value	—	1	1	1	1	1	1	1
PV-F5	Mean	1.78E−02	7.63E−01ǂ	**4.31E−03** ξ	3.92E−01ǂ	9.80E−02ǂ	4.41E−02ǂ	7.17E−02ǂ	1.44E + 00ǂ
	Std	4.26E−03	2.41E−01	**1.91E−02**	5.00E−01	1.14E−01	5.57E−03	1.54E−02	3.83E−01
	P value	—	5.58E−14	2.40E−03	9.71E−04	1.87E−03	4.39E−14	2.68E−15	9.40E−16
	H value	—	1	1	1	1	1	1	1
ǂ/ξ/~		—	5/0/0	3/2/0	5/0/0	4/0/1	4/0/1	5/0/0	5/0/0

silicon R.T.C. France solar cell under 1000 W/m^2 at 33°C. Three different PV modules are used to test the performance of ASDE, i.e., ploycrystalline Photowatt-PWP201 (PV–F3), monocrystalline STM6-40/36 (PV–F4), and polycrystalline STP6-120/36 (PV–F5). The ploycrystalline Photowatt-PWP201 is gauged under 1000 W/m^2 at 45°C [62]. The monocrystalline STM6-40/36 and the polycrystalline STP6-120/36 are gauged under 51°C and 55°C, respectively, whose current-voltage data are obtained from [63]. The feasible range of parameters to be extracted is displayed in Table 5. In addition, the parameter setting of the simulation is the same to Section 4.

5.3. Comparison on Solution Accuracy. The basic settings of Table 6 are the same to Table 1. It can be seen clearly from Table 4 that ASDE obtains the best performance on 2 photovoltaic models (PV–F1 and PV-F2) and gets the second performance on the other models. Meanwhile, ASDE is better than DE, PSO, AMFO, and RLDE on all five photovoltaic models. Compared with SLPSO and MFO, ASDE is better on four models and is similar on one model. In addition, ASDE has obtained the better performance than EPSDE on three models. This indicates that ASDE is significant in solving the problem of parameter extraction of the photovoltaic model. Finally, ASDE is worse than EPSDE on 2 models (PV–F4 and PV-F5), which may be because the ε-greedy strategy (ε = 0.4) is slightly poor in the exploitation ability of the population.

It is noteworthy that the root mean square error between the current obtained by ASDE and the theoretical measured current is less than 0.05. Thus, we believe that ASDE is effective in optimizing the parameter extraction problem of photovoltaic models.

5.4. The Comparison Results of Convergence Speed. In Figure 6, the vertical axis is the nature logarithm of the mean value over independent 25 runs, and the horizontal axis is the sampling point where sampling points are taken from FES = 1000 and mod (FES, 10000) = 0.

Figure 6 shows that ASDE gains the best convergence speed on 3 models (PV–F1, PV-F2, and PV-F3), which suggests that ASDE can get the promising convergence performance in optimizing the parameter extract problems of photovoltaic models. On model PV-F4, in the early stage of search, the convergence speed of ASDE is better than EPSDE. However, in the later stage of the search, ASDE

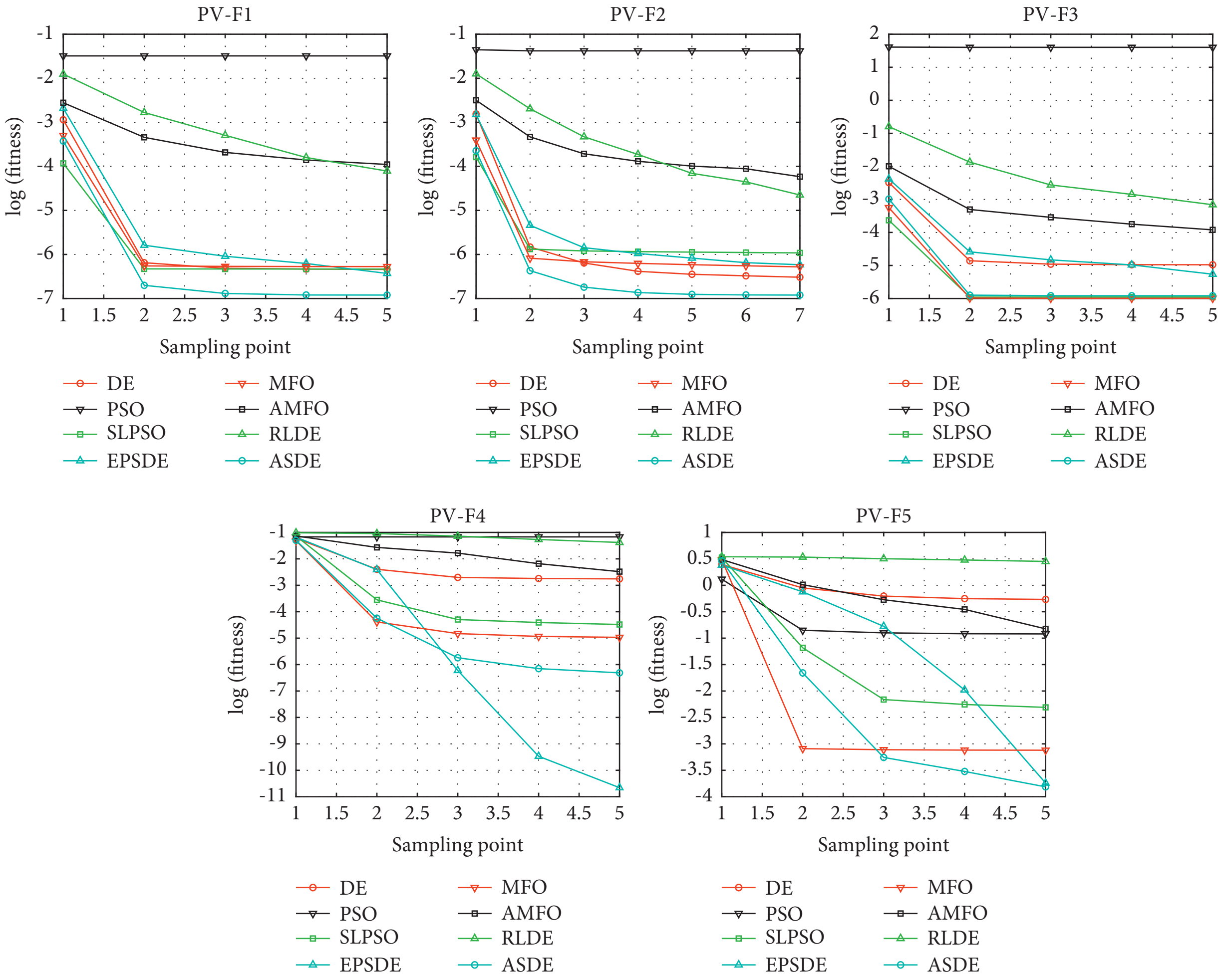


Figure 6: Convergence performance of the eight compared algorithms on 5 PV models.

obtains worse convergence speed than EPSDE. This could be because ASDE with fixed ε value has the slightly poor convergence ability, which is for avoiding the population falling into the trap of local optimal solution. Although ASDE gains worse convergence performance than MFO on model PV-F5, it achieves the best convergence accuracy. This indicates that the exploration ability of the ASDE is interesting.

6. Conclusions

In reality, most optimization problems are black box optimization problems with unknown problem features, which makes the traditional adaptive DE algorithm unable to achieve satisfactory performance in optimizing those optimization problems. To improve this situation, a DE with autonomous selection of mutation strategies and control parameters is proposed in this paper, named ASDE for short. In ASDE, the historical experience archive with population features is utilized to preserve historical experience of mutation strategies and control parameters, which makes the historical experience can be cumulated so that the accumulated historical experience can be mapped into rules repository, and the individuals can choose the combination of mutation strategies and control parameters according to those rules. In addition, to assure that the historical experience can be effectively accumulated and utilized efficiently, an updating and utilization mechanism of the historical experience is proposed in this paper. Finally, the 15 functions of CEC2015 and the parameter extraction problems of the photovoltaic model are utilized to verify the performance of ASDE, and the simulation results demonstrate that ASDE outperforms seven compared algorithms.

In our future work, ASDE will be applied to solve other real-world optimization problems to further test its performance.

Data Availability

The data used to support the findings of this study cannot be shared due to privacy.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Zhenyu Wang and Zijian Cao were responsible for conceptualization and methodology. Investigation was conducted by Chen Liu and Haowen Jia. Writing-original draft preparation was carried out by Zhenyu Wang. Zijian Cao and Feng Tian were responsible for writing-review and editing. Binhui Han and Fuxi Liu were responsible for funding acquisition. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research was partially funded by the Shaanxi Natural Science Basic Research Project (grant no. 2020JM-565).

References

[1] R. M. Storn and K. Price, "Differential Evolution-A simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 23, 1995.

[2] K. K. Mandal and N. Chakraborty, "Parameter study of differential evolution based optimal scheduling of hydrothermal systems," *Journal of hydro-environment research*, vol. 7, no. 1, pp. 72–80, 2013.

[3] P. Kumar, M. Pant, and A. Abraham, "Two enhanced differential evolution variants for solving global optimization problems," in *Proceedings of the 2011 Third World Congress on Nature and Biologically Inspired Computing*, pp. 201–206, Salamanca, Spain, October 2011.

[4] D. O. Silva, L. G. M. Vieira, F. S. Lobato, and M Barrozo, "Optimization of the design and performance of hydrocyclones by Differential Evolution technique," *Chemical Engineering and Processing: Process Intensification*, vol. 61, pp. 1–7, 2012.

[5] W. F. Sacco, A. A. de Moura Meneses, and N. Henderson, "Some studies on differential evolution variants for application to nuclear reactor core design," *Progress in Nuclear Energy*, vol. 63, pp. 49–56, 2013.

[6] R. Arya and S. C. Choube, "Differential evolution based technique for reliability design of meshed electrical distribution systems," *International Journal of Electrical Power & Energy Systems*, vol. 48, pp. 10–20, 2013.

[7] L. d. S. Coelho, V. C. Mariani, M. V. Ferreira da Luz, and J. V Leite, "Novel gamma differential evolution approach for multiobjective transformer design optimization," *IEEE Transactions on Magnetics*, vol. 49, no. 5, pp. 2121–2124, 2013.

[8] G. Li, Y. Li, H. Chen, and W. Deng, "Fractional-order controller for cours*E*–keeping of underactuated surface vessels based on frequency domain specification and improved particle swarm optimization algorithm," *Applied Sciences*, vol. 12, no. 6, p. 3139, 2022.

[9] W. Deng, Z. Li, X. Li, H. Chen, and H Zhao, "Compound fault diagnosis using optimized MCKD and sparse representation for rolling bearings," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–9, 2022.

[10] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1430–1442, 2021.

[11] F. Zhao, L. Zhao, L. Wang, and H Song, "An ensemble discrete differential evolution for the distributed blocking flowshop scheduling with minimizing makespan criterion," *Expert Systems with Applications*, vol. 160, Article ID 113678, 2020.

[12] M. G. H. Omran and A. P. Engelbrecht, "Self-adaptive differential evolution method for unsupervised image classification," in *Proceedings of the 2006 IEEE Conference on Cybernetics and Intelligent Systems*, pp. 1–6, Bangkok, Thailand, June 2006.

[13] E. Cuevas, D. Zaldivar, and M. Pérez-Cisneros, "A novel multi-threshold segmentation approach based on differential evolution optimization," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5265–5271, 2010.

[14] A. Nakib, B. Daachi, and P. Siarry, "Hybrid differential evolution using low-discrepancy sequences for image segmentation," in *Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, pp. 634–640, Shanghai, China, May 2012.

[15] P. Mesejo, R. Ugolotti, F. Di Cunto, M. Giacobini, and S Cagnoni, "Automatic hippocampus localization in histological images using Differential Evolution-based deformable models," *Pattern Recognition Letters*, vol. 34, no. 3, pp. 299–307, 2013.

[16] J. X. Du, D. S. Huang, X. F. Wang, and X Gu, "Shape recognition based on neural networks trained by differential evolution algorithm," *Neurocomputing*, vol. 70, no. 4-6, pp. 896–903, 2007.

[17] E. N. Dragoi, S. Curteanu, A. I. Galaction, and D Cascaval, "Optimization methodology based on neural networks and self-adaptive differential evolution algorithm applied to an aerobic fermentation process," *Applied Soft Computing*, vol. 13, no. 1, pp. 222–238, 2013.

[18] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Differential evolution mutations: taxonomy, comparison and convergence analysis," *IEEE Access*, vol. 9, Article ID 68629, 2021.

[19] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution[C]//2005 IEEE congress on evolutionary computation," *IEEE*, vol. 1, pp. 506–513, 2005.

[20] A. K. Mohamed and A. W. Mohamed, "Real-parameter unconstrained optimization based on enhanced AGDE algorithm," *Machine Learning Paradigms: Theory and Application*, Springer, Cham, pp. 431–450, 2019.

[21] L. Cui, G. Li, Z. Zhu et al., "Adaptive multipl*E*–elites-guided composite differential evolution algorithm with a shift mechanism," *Information Sciences*, vol. 422, pp. 122–143, 2018.

[22] A. W. Mohamed and A. K. Mohamed, "Adaptive guided differential evolution algorithm with novel mutation for numerical optimization," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 2, pp. 253–277, 2019.

[23] Z. Zhao, J. Yang, Z. Hu, and H Che, "A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems," *European Journal of Operational Research*, vol. 250, no. 1, pp. 30–45, 2016.

[24] A. Ghosh, S. Das, A. Chowdhury, and R Giri, "An improved differential evolution algorithm with fitness-based adaptation of the control parameters," *Information Sciences*, vol. 181, no. 18, pp. 3749–3765, 2011.

[25] Z. Meng, J. S. Pan, and L. Kong, "Parameters with adaptive learning mechanism (PALM) for the enhancement of

differential evolution," *KnowledgE–Based Systems*, vol. 141, pp. 92–112, 2018.

[26] J. Brest, S. Greiner, B. Boskovic, 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.

[27] A. Zamuda and J. Brest, "Self-adaptive control parameters' randomization frequency and propagations in differential evolution," *Swarm and Evolutionary Computation*, vol. 25, pp. 72–99, 2015.

[28] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive[J]," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[29] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proceedings of the 2013 IEEE congress on evolutionary computation*, pp. 71–78, Cancun, Mexico, June 2013.

[30] X. Yu, M. Cai, and J. Cao, "A novel mutation differential evolution for global optimization," *Journal of Intelligent and Fuzzy Systems*, vol. 28, no. 3, pp. 1047–1060, 2015.

[31] R. Mallipeddi and P. N. Suganthan, "Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies," *Swarm, Evolutionary, and Memetic Computing, Proceedings of the International conference on swarm, evolutionary, and memetic computing*, Springer, Applied Soft Computing, vol. 11, no. 2, pp. 1679–1696, Berlin, Heidelberg, 2011.

[32] L. Cui, G. Li, Q. Lin, J. Chen, and N Lu, "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations," *Computers & Operations Research*, vol. 67, pp. 155–173, 2016.

[33] 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.

[34] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang, and H Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 329, pp. 329–345, 2016.

[35] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P Suganthan, "Ensemble of differential evolution variants," *Information Sciences*, vol. 423, pp. 172–186, 2018.

[36] S. Das, A. Abraham, U. K. Chakraborty, and A Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.

[37] Y. Cai and J. Wang, "Differential evolution with neighborhood and direction information for numerical optimization," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2202–2215, 2013.

[38] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 99–119, 2011.

[39] Y. Wang, H. X. Li, T. Huang, and L Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Applied Soft Computing*, vol. 18, pp. 232–247, 2014.

[40] Y. Wang, Z. Z. Liu, J. Li, H. X. Li, and G. G. Yen, "Utilizing cumulative population distribution information in differential evolution," *Applied Soft Computing*, vol. 48, pp. 329–346, 2016.

[41] Z. H. Zhan, X. F. Liu, H. Zhang et al., "Cloudde: a heterogeneous differential evolution algorithm and its distributed cloud version," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 3, pp. 704–716, 2017.

[42] Z. H. Zhan, Z. J. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Transactions on Cybernetics*, vol. 50, no. 11, pp. 4633–4647, 2020.

[43] S. Hassan, A. M. Hemeida, S. Alkhalaf, A. A. Mohamed, and T. Senjyu, "Multi-variant differential evolution algorithm for feature selection," *Scientific Reports*, vol. 10, no. 1, Article ID 17261, 2020.

[44] R. Storn, "On the usage of differential evolution for function optimization," in *Proceedings of the Proceedings of North American Fuzzy Information Processing*, pp. 519–523, Berkeley, CA, USA, June 1996.

[45] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," *Lecture Notes in Computer Science*, in *Proceedings of the Australasian joint conference on artificial intelligence*, pp. 861–872, Berlin, Heidelberg, December 2004.

[46] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization," vol. 29Technical Report201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, pp. 625–640, Nanyang Technological University, Singapore, 2014.

[47] J. Kennedy and R. Eberhart, "Particle swarm optimization [C]//Proceedings of ICNN'95-international conference on neural networks," *IEEE*, vol. 4, pp. 1942–1948, 1995.

[48] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Information Sciences*, vol. 291, pp. 43–60, 2015.

[49] S. Mirjalili, "Moth-flame optimization algorithm: a novel naturE–inspired heuristic paradigm," *KnowledgE–Based Systems*, vol. 89, pp. 228–249, 2015.

[50] X. Zhao, Y. Fang, Z. Ma, and M Xu, "An Ameliorated moth-flame optimization algorithm," in *Proceedings of the 2018 37th Chinese Control Conference (CCC)*, pp. 2372–2377, Wuhan, China, July 2018.

[51] Z. Hu, W. Gong, and S. Li, "Reinforcement learning-based differential evolution for parameters extraction of photovoltaic models," *Energy Reports*, vol. 7, pp. 916–928, 2021.

[52] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization,"Zhengzhou, China, Technical Report 2017.

[53] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: algorithm jSO," in *Proceedings of the 2017 IEEE congress on evolutionary computation (CEC)*, pp. 1311–1318, Donostia, Spain, June 2017.

[54] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M Jambi, "LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems," in *Proceedings of the 2017 IEEE Congress on evolutionary computation (CEC)*, pp. 145–152, Donostia, Spain, June 2017.

[55] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, "Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization," *Swarm and Evolutionary Computation*, vol. 50, Article ID 100455, 2019.

[56] S. Xu and Y. Wang, “Parameter estimation of photovoltaic modules using a hybrid flower pollination algorithm,” *Energy Conversion and Management*, vol. 144, pp. 53–68, 2017.

[57] W. C. Yeh, C. L. Huang, P. Lin, Z. Chen, Y. Jiang, and B Sun, “Simplex simplified swarm optimisation for the efficient optimisation of parameter identification for solar cell models,” *IET Renewable Power Generation*, vol. 12, no. 1, pp. 45–51, 2018.

[58] A. Askarzadeh and A. Rezazadeh, “Parameter identification for solar cell models using harmony search-based algorithms,” *Solar Energy*, vol. 86, no. 11, pp. 3241–3249, 2012.

[59] M. R. AlRashidi, M. F. AlHajri, K. M. El-Naggar, and A Al-Othman, “A new estimation approach for determining the I-V characteristics of solar cells,” *Solar Energy*, vol. 85, no. 7, pp. 1543–1550, 2011.

[60] D. Oliva, M. Abd El Aziz, and A. Ella Hassanien, “Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm,” *Applied Energy*, vol. 200, pp. 141–154, 2017.

[61] S. Li, W. Gong, X. Yan et al., “Parameter extraction of photovoltaic models using an improved teaching-learning-based optimization,” *Energy Conversion and Management*, vol. 186, pp. 293–305, 2019.

[62] T. Easwarakhanthan, J. Bottin, I. Bouhouch, and C Boutrit, “Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers,” *International Journal of Solar Energy*, vol. 4, no. 1, pp. 1–12, 1986.

[63] N. T. Tong and W. Pora, “A parameter extraction technique exploiting intrinsic properties of solar cells,” *Applied Energy*, vol. 176, pp. 104–115, 2016.