Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4544818, 18 pages
https://doi.org/10.1155/2022/4544818

Research Article

An External Selection Mechanism for Differential Evolution Algorithm

Haigang Zhang and Da Wang

School of Software, Yunnan University, Kunming 650000, China

Correspondence should be addressed to Da Wang; wangda@mail.ynu.edu.cn

Received 15 September 2021; Revised 16 February 2022; Accepted 14 March 2022; Published 4 April 2022

Academic Editor: Miaolei Zhou

The procedures of differential evolution algorithm can be summarized as population initialization, mutation, crossover, and selection. However, successful solutions generated by each iteration have not been fully utilized to our best knowledge. In this study, an external selection mechanism (ESM) is presented to improve differential evolution (DE) algorithm performance. The proposed method stores successful solutions of each iteration into an archive. When the individual is in a state of stagnation, the parents for mutation operation are selected from the archive to restore the algorithm's search. Most significant of all, a crowding entropy diversity measurement in fitness landscape is proposed, cooperated with fitness rank, to preserve the diversity and superiority of the archive. The ESM can be integrated into existing algorithms to improve the algorithm's ability to escape the situation of stagnation. CEC2017 benchmark functions are used for verification of the proposed mechanism's performance. Experimental results show that the ESM is universal, which can improve the accuracy of DE and its variant algorithms simultaneously.

1. Introduction

Differential evolution (DE) algorithm [1], introduced by Storn and Price in 1995, is one of the most popular evolutionary algorithms. DE is easy to be understood, and the principle is simple. Still, it demonstrates good optimization ability and is used in many single-objective optimization problems successfully, including continuous optimization [2], discrete optimization [3], constrained optimization [4], and unconstrained optimization [5].

Different from other meta-heuristic algorithms [6], the procedures of differential evolution algorithms can be summarized as population initialization, mutation, crossover, and selection. After each iteration, promising solutions with better fitness values are selected to survive to the next iteration. These solutions can be called successful solutions. DE repeats this procedure until a predefined termination criterion is reached. However, DE may suffer stagnation during the evolution process, which means stop generating successful solutions and converging to a fixed point [7]. When the population stagnates, taking appropriate strategies to restore the search should theoretically further improve the algorithm's performance.

Successful solutions are generally located in valleys of the fitness landscape. As the iteration progresses, successful solutions containing useful information may be discarded and underutilized. We, therefore, propose an external selection mechanism (ESM). First, ESM stores successful solutions of each iteration into an archive. When the individual is in a state of stagnation, the parents for mutation operation are selected from the archive. In this way, the population can regain diversity and restore searchability. Second, the diversity and superiority of the archive are relatively significant because they can directly affect the performance of the algorithm. Therefore, a rule is proposed to update the archive based on novel crowding entropy diversity measurement in the fitness landscape and fitness rank.

To verify and analyze the effectiveness of the ESM, we performed a series of experiments and comparisons on the

CEC2017 benchmark set [8], incorporating three classic DEs and five state-of-the-art DE variants. Results indicate that the ESM can effectively improve the algorithm performance without increasing the computational complexity.

The rest of this study is organized as follows: Section 2 introduces the canonical DE algorithm, including its typical mutation, crossover, and selection operators. Section 3 reviews related work. The proposed ESM procedures are introduced in Section 4. The effectiveness of the proposed mechanism is discussed in Section 5 based on the computational results. Conclusions and future work are illustrated in Section 6.

2. Differential Evolution

In this section, we introduce the basic differential evolution algorithm, including the well-known mutation strategy DE/rand/1 [1] and other widely used mutation strategies.

2.1. Initialization. An initial random population P consists of NP individuals, each represented by $\{X_i^t = (x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t) | i = 1, 2, \ldots, \mathrm{NP}\}$, where $t = 0, 1, 2, \ldots, T_{\max}$ is the iteration number, and D is the number of dimensions in the solution space. In DE, uniformly distributed random functions are used to generate initial solutions $x_{i,j}^0 = x_{j,\min} + \mathrm{rand}(0,1) \cdot (x_{j,\max} - x_{j,\min})$, where $x_{j,\max}$ and $x_{j,\min}$ are the maximum and minimum boundary values, respectively, on the corresponding jth dimension.

2.2. Mutation. At iteration t, for each target vector X_i^t, a mutant vector V_i^t is generated according to the following:

$$\mathrm{DE/rand/1}: V_i^t = X_{r_1}^t + F \cdot \left(X_{r_2}^t - X_{r_3}^t\right). \tag{1}$$

Other widely used mutations strategies include

$$\begin{aligned}
\frac{\mathrm{DE}}{(\mathrm{best/1})}&: V_i^t = X_{\mathrm{best}}^t + F \cdot \left(X_{r_1}^t - X_{r_2}^t\right),\\
\frac{\mathrm{DE}}{(\mathrm{best/2})}&: V_i^t = X_{\mathrm{best}}^t + F \cdot \left(X_{r_1}^t - X_{r_2}^t\right) + F \cdot \left(X_{r_3}^t - X_{r_4}^t\right),\\
\frac{\mathrm{DE}}{(\mathrm{rand/2})}&: V_i^t = X_{r_1}^t + F \cdot \left(X_{r_2}^t - X_{r_3}^t\right) + F \cdot \left(X_{r_4}^t - X_{r_5}^t\right),\\
\frac{\mathrm{DE}}{(\mathrm{current-to-best/1})}&: V_i^t = X_i^t + F \cdot \left(X_{\mathrm{best}}^t - X_i^t\right) + F \cdot \left(X_{r_1}^t - X_{r_2}^t\right),\\
\frac{\mathrm{DE}}{(\mathrm{current-to-rand/1})}&: V_i^t = X_i^t + F \cdot \left(X_{r_1}^t - X_i^t\right) + F \cdot \left(X_{r_2}^t - X_{r_3}^t\right),
\end{aligned} \tag{2}$$

where r_1, r_2, r_3, r_4, and r_5 are mutually different integers randomly generated from the range $(1, 2, \ldots, \mathrm{NP})$, and they are different from i $(r_1 \neq r_2 \neq r_3 \neq i)$. X_{best}^t is the individual vector with the best fitness value in the population at iteration t.

2.3. Crossover. We illustrate the binomial crossover, in which the target vector X_i^t and donor vector V_i^t exchange elements according to the following rules:

$$u_{i,j}^t = \begin{cases} v_{i,j}^t, & \text{if } \left(\mathrm{rand}(0,1) < \mathrm{CR} \text{ or } j = j_{\mathrm{rand}}\right),\\ x_{i,j}^t, & \text{otherwise,}\end{cases} \tag{3}$$

where the crossover rate $\mathrm{CR} \in [0,1]$ is a uniformly distributed random integer in $[1, D]$ that ensures at least one element of the trial vector is inherited from the donor vector.

2.4. Selection. The greedy selection strategy is generally used in DE. The variable, X_i^{t+1}, is assigned when the fitness value of the trial vector, U_i^t, is equal to or better than X_i^t, and otherwise X_i^t is reserved:

$$X_i^{t+1} = \begin{cases} U_i^t, & f\left(U_i^t\right) \leq f\left(X_i^t\right),\\ X_i^t, & \text{otherwise,}\end{cases} \tag{4}$$

where $f(x)$ is the fitness function.

3. Related Work

Most research on improving DE has focused on mutation operator [9]. In recent decades, there have been many mutation operators with distinct search performance that have been proposed. Fan et al. [10] proposed a triangular mutation strategy, which was considered a local search operator. Zhang et al. [11] introduced a new DE variant, JADE, improving optimization performance by a new mutation strategy, DE/current − to − $p$best/1. DE/current − to − $p$best/1 is one of the most successful mutation operators because of its relatively balanced performance between exploration and exploitation. Wang et al. [12] proposed a new mutation strategy called DE/current − to − $l$best/1 based on the values near the current parameter to keep balance of exploitation and exploration capabilities during the differential evolution. A novel DE algorithm with intersect mutation operation called intersect mutation differential evolution (IMDE) was proposed [13] to further improve the performance of the standard DE algorithm. Mohamed et al. [14] proposed two mutation strategies, DE/current – to – ord_best/1 and DE/current – to – ord_*p*best/1. Both of the proposed mutation strategies were based on ordering three selected vectors to achieve different search behaviors. Deng et al. [15] proposed a novel DE variant called DCDE based on a dynamic combination-based mutation operator to achieve an appropriate balance between global exploration ability and local exploitation ability.

In addition, novel neighborhood-based [16], dimension-based [17], opposition-based [18], and regeneration-based [19] mechanisms also showed their effectiveness on improving the performance of DE and its variants. In this study, we focus on the related work on archive-based techniques for DE. In the multiobjective evolutionary algorithm, a subpopulation called the external archive is used to store nondominated solutions that have been found during the search [20, 21], employing an archive is almost standard in multiobjective optimization [22]. The goal of single-objective global numerical optimization is to find a global optimum in decision space of a given objective function. Recently, archive is also introduced into global numerical optimization. In JADE [11], a set of recently explored inferior solutions are archived, and their difference from the current population is considered as a promising direction toward the optimum. The archive is applied to the second difference vector of the mutation strategy in JADE. Zhou et al. [23] proposed a DE framework with guiding archive (GAR-DE), GAR-DE chose the base vector of mutation strategy from the archive to help DE escape from the situation of stagnation, and Manhattan distance metric was used to maintain the diversity of the archive. Zeng et al. [24] proposed a new selection operator (NSO) for DE, which archived not only the successful updated solutions but also the discarded trial solutions. The NSO focused on selecting appropriate solution from the archive to survive to the next generation.

As can be seen from preceding explanation, the differences among archive-based mechanisms mainly consist in two aspects: what information/solutions should be stored in the archive; how to maintain and update the archive; and how to use the archive to guide the search. The most striking feature of the proposed ESM is the use of information entropy rather than distance metric [23, 25] to maintain the diversity of the archive.

Shannon defined the information entropy theory [26] for the first time in 1948. In information theory, entropy is used to measure the expected value of a random variable. As the evaluation scale of information quantity of stochastic process, information entropy has been extended to a general and effective tool to solve difficult numerical problems and uncertain polynomial combinatorial optimization problems [27, 28]. An information theoretic technique was adopted to analyze the ruggedness of a continuous fitness landscape in [29, 30]. Petalas et al. [27] proposed a memetic particle swarm optimization algorithm that exploits Shannon's information entropy for decision-making in swarm level, as well as a probabilistic decision-making scheme in particle level, for determining when and where local search is applied. As can be seen from the previous studies, entropy actually reflects the degree of chaos of system. Therefore, entropy principle can be a promising method applied to evolutionary algorithm to ensure population diversity.

4. The Proposed ESM

In this section, we discuss the characteristics of the successful solution archive and archive-updating procedure and describe the ESM implementation.

4.1. Successful Solution Archive. From Section 2, we can know that DE selects one vector from the trial vector and parent vector to survive to the next generation, and the survivor is known as the successful solution. As shown in Figure 1(a), in the two-dimensional solution space of a multimodal problem, the population is evenly distributed at the initial stage. As the iteration continues, from Figure 1(b) we can conclude that superior successful solutions (red points) can be generated in possible optimal regions. However, in Figure 1(c), the random and greedy selection operation makes the population gradually converge to a local optimum. Some successful solutions may be discarded without being fully exploited because of the greedy selection operation as the iteration progresses. An appropriate mechanism to utilize the successful solutions with high diversity contribution can make the algorithm reexplore the missed potential optimal solution. To some extent, it is also a remedy for the greedy selection operation.

Considering the above analysis, as depicted in Figure 2, an external archive is designed to store successful solutions generated during each iteration. When an individual is in a state of stagnation, the archive is activated for parent selection of mutation operation. For instance, the classic mutation strategy DE/rand/1 can be modified to

$$V_i^t = X_{ar_1}^t + F \cdot \left(X_{ar_2}^t - X_{ar_3}^t \right), i\varepsilon I, \tag{5}$$

where $ar_1, ar_2, ar_3 \in A^t \wedge ar_1 \neq ar_2 \neq ar_3$, A^t is the successful solution archive at iteration t. A simple but efficient method is adopted at this point, when the number of steps for the individual's fitness value to stop updating reaches a predetermined value Q, the individual is considered in a stagnant state [31].

4.2. Archive Updating Procedure. As the successful solution archive has been established, the current problem is how to update the archive to keep its diversity and superiority in the whole evolution process. An archive containing diverse successful solutions can drag the individual out of stagnation without crippling the algorithm's performance.

4.2.1. Individual Diversity Measure. To our best knowledge, in single-objective optimization, diversity is usually defined by the spatial distribution of the whole population [32], but there is few diversity contribution measures for a single

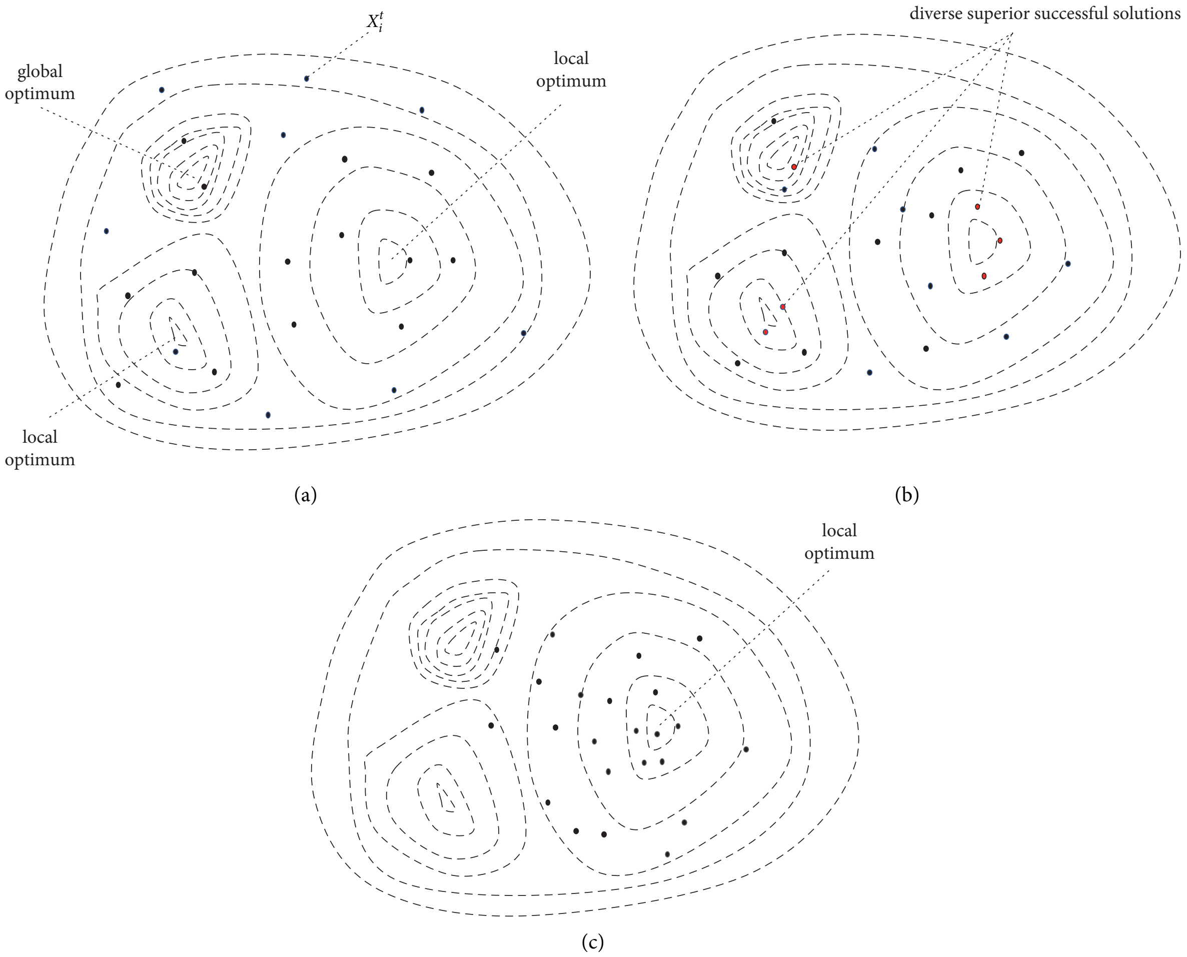


FIGURE 1: Illustration of the successful solution distribution in a two-dimensional space: (a) at the initial stage; (b) at the intermediate stage; (c) at the terminal stage.

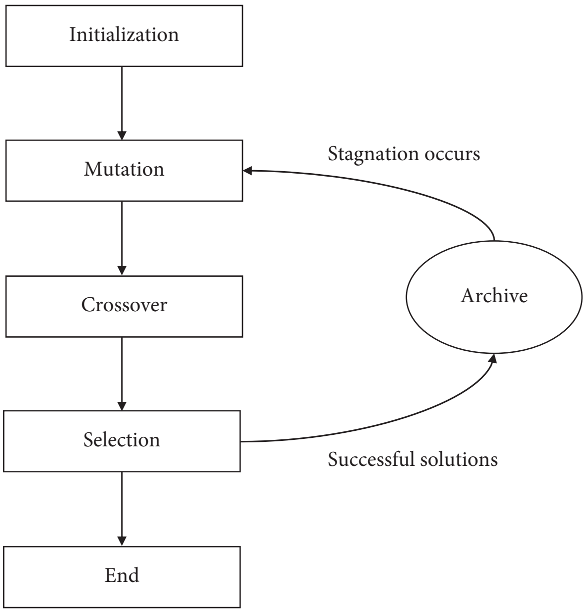


FIGURE 2: Illustration of the successful solution archive.

individual to whole population. Wang et al. [33] proposed an Euclidean distance-based diversity metric for each individual, but a huge computational cost is needed to calculate the distance between every two individuals. To utilize diversity information while reducing the computational cost, we can estimate the diversity by fitness value distribution on individual level. Fitness distance can represent the spatial distance to some extent [34].

In multiobjective optimization, Deb et al. [35] proposed a crowding distance measure method to get an estimation of the density of solutions surrounding a particular solution in the population, by calculating the average distance of two points on either side of this point along each of the objectives. However, the distribution of this point is not considered, and the only average distance may not accurately reflect the crowding degree. With utilizing the characteristic of fitness distance, we can extend the crowding distance measure to single-objective optimization. For example, in Figure 3, the crowding distance of the solution A can be calculated as $3L + 1L = 4L$ (according to crowding distance calculation method in [35]), which is the same as the solution B $(2L + 2L = 4L)$, whereas solution B obviously has a higher diversity contribution than solution A.

4.2.2. Archive Update Procedure. From the above statement, this study proposes a crowding entropy diversity measurement in the fitness landscape at individual level. As shown in Figure 4, we sort the archive population in the ascending order of fitness during each iteration, then, a fitness distance estimation operator can be formulated as follows:

$$Dl(i) = |f_{i+1} - f_i|, \quad (6)$$

$$Du(i) = |f_i - f_{i-1}|, \quad (7)$$

where f_i is the fitness value of the i th individual. $Dl(i)$ and $Du(i)$ can be further normalized, then, the crowding entropy of the i th individual can be defined as follows:

$$P_{i1} = \frac{Dl(i)}{Dl(i) + Du(i)}, \quad (8)$$

$$P_{i2} = \frac{Du(i)}{Dl(i) + Du(i)}, \quad (9)$$

$$\mathrm{CE}_i = -\sum_{j=1}^{2} P_{ij}\log_2\left(P_{ij}\right), \quad (10)$$

where $\mathrm{CE}_i \in [0, 1]$, the larger CE means that the individual distribution is more uniform, so it has a higher diversity contribution. According to Figure 3, we can further calculate $\mathrm{CE}_A = 0.8$ and $\mathrm{CE}_B = 1$, this is also consistent with their diversity contribution. Since the best and worst individuals locate in the boundary scope, there is only one neighbor around them, respectively. Therefore, we set $\mathrm{CE}_1 = \mathrm{CE}_2$ and $\mathrm{CE}_{\mathrm{NP}} = \mathrm{CE}_{\mathrm{NP}-1}$ for the ESM to function normally. To keep the superiority of the archive, fitness value rank is further introduced:

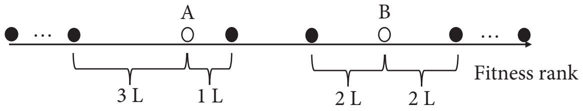


Figure 3: Crowding distance presented in fitness landscape.

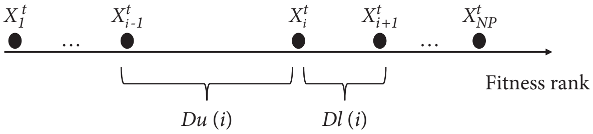


Figure 4: Crowding entropy presented in fitness landscape.

$$\mathrm{FR}_i = i, \quad i = 1, 2, \ldots, \mathrm{NP}, \quad (11)$$

$$R_i = \left(1 - \frac{\mathrm{FR}_i}{\mathrm{NP} + 1}\right) \cdot \mathrm{CE}_i, \quad (12)$$

where R_i represents the priority of the individual to be updated. An individual in the archive with a higher rank (i.e., far from the global optimum) or with lower entropy (i.e., located in a dense area) should have a higher probability of being updated.

4.3. The Implementation of the ESM. Based on the ESM described above, we present the ESM updating procedure in Algorithm 1 and the complete implementation based on classic DE with mutation strategy DE/rand/1 in Algorithm 2. Different from the existing alternative for the selection of parents [33, 36], the proposed mechanism is an adaptive approach that considers the feedback of the recently successful solutions so that the robustness of the DE algorithm has the potential to be enhanced by dynamically adapting promising parents for the situation of stagnation. Moreover, the proposed ESM is computationally efficient. The time complexities of updating operations of the archive are $O(\mathrm{NP})$. Additionally, the selection of parents, counting of the consecutive unsuccessful updates, and stagnation detection do not increase the overall time complexity.

5. Experiments and Comparisons

In this section, the proposed ESM is first integrated into three representative classic DE algorithms, including DE/rand/1, DE/best/1, and DE/current − to − best/1, to verify its efficiency. Then, five state-of-the-art DE variants, including j2020 [37], EJADE [38], BeSD [39], EBSHADE [14], and EBLSHADE [14], are introduced to incorporate with the ESM for further comparison. Finally, all experiments are conducted on the CEC2017 benchmark set [8], which contains 30 representative benchmark functions in four categories: unimodal (F1–F3), simple multimodal (F4–F10), hybrid (F11–F20), and composition (F21–F30). This benchmark set is widely used in the performance testing of evolutionary algorithms [9]. It is worth mentioning that the ESM does not change existing mechanisms in algorithms

```
Input: successful solution $X_i$, the archive $A$;
Output: the archive $A$;
FOR $i = 1$ to NP
        Calculate $R_i$ of $A_i$ by equations (6)–(12);
END FOR
Get the index $j$ of the smallest $R$;
Replace the solution $A_j$ with $X_i$;
```

Algorithm 1: The pseudo code of ESM updating procedure.

```
Step 1: Initialization
Set iteration $t = 0$, fitness evaluations FE$s$ = 0, stagnation threshold $Q = 120$.
Initialize a population of NP individuals $P^0 = \{X_1^0, \ldots, X_{NP}^0\}$ with
$X_i^0 = \{x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t\}$;
Set parameter $F = 0.5$ and CR = 0.5, stagnation iterations $q_1^0 = \cdots = q_{\mathrm{NP}}^0 = 0$, archive
$A^0 = P^0$;
Step 2: Evolution Iteration
WHILE FE$s$ < Max_FE$s$
DO
        Step 2.1: Mutation Operation
        FOR $i = 1$ to NP
                Randomly select $r_1 \neq r_2 \neq r_3$ from the set $I = \{1, 2, \ldots, NP\}$;
                IF $q_i^t < Q$
                        Generate donor vectors $V_i^t$ by (1);
                ELSE
                        Generate donor vectors $V_i^t$ by (5);
                END
        END FOR
        Step 2.2: Crossover Operation
        FOR $i = 1$ to NP
                Generate trial vectors $U_i^t$ by (3);
        END FOR
        Step 2.3: Selection and Archive Update
        FOR $i = 1$ to NP
                IF $f(U_i^t) < f(X_i^t)$
                        $q_i^{t+1} = 0$;
                        Update $A$ according to Algorithm 1 with $U_i^t$;
                ELSE
                        $q_i^{t+1} = q_i^t + 1$;
                END IF
                Generate new vectors $X_i^{t+1}$ by (4);
        END FOR
        Step 2.4: Increase the Count
        $FEs = FEs + NP$;
        $t = t + 1$
END WHILE
```

Algorithm 2: The pseudo code of DE/rand/1 with the proposed ESM.

including the bound constraint handling and linear population size reduction.

The parameters of all algorithms are listed in Table 1. The common parameters of the incorporated algorithms are set as follows: The population size NP is 100, the stagnation tolerance Q is 120, and the maximum allowed number of function evaluations Max_FEs = $10000 * D$, where D is the problem dimension. Moreover, H represents the historical memory size associated with the adaptive control of scaling factors (F) and crossover rates (CR), LP is the learning period.

Limited by the article space, this study presents results at $D = 30$. For the fairness of comparison, each algorithm was implemented in MATLAB 2017b and executed over 51 independent runs on a Windows 10 64 bit desktop PC with 32 GB of RAM and a 3.0 GHz Intel Core i7-9700 processor.

5.1. Improving the Performance of Classic DEs and State-of-the-Art DE Variants.

The following tables show the mean value (Mean) and standard deviation (Std.Dev) of the error value $f(X) - f(X^*)$, where $f(X^*)$ representing the global optimum. The "+," "−," and "=" signs at each row of the table indicates that the ESM-based algorithm is, respectively, better than, equal to, or worse than the comparison algorithm.

Tables 2–4 show the performance of three classic DEs. For DE/rand/1, which is widely recognized for its diversity maintaining capability and relative low search efficiency, the ESM significantly improves its performance on 24 test functions and only cause performance degradation on two test functions, as shown in Table 2. DE/best/1 is well known as its local exploitation capability and fast convergence speed, the ESM also makes progress on 19 functions, as presented in Table 3. Relatively, the performance of ESM-DE/best/1 decreases on five functions including F13, F16, F24, F27, and F29 and keeps the same as DE/best/1 on the rest six functions. DE/current − to − best/1 is more inclined to balance global exploration and local exploitation. From Table 3, we can find that the ESM also achieves improvement on 26 functions. The results of ESM-DE/current − to − best/1 only decrease on two functions including F2 and F19. With functions F25 and F27, the two competitors keep the same.

From the above experiments, we can conclude that the ESM can significantly improve classic DE's performance. To further verify the ESM's universality, five representative and advanced DE variants are introduced, Tables 5–9 show these variants and their competitors' performance. For ESM-j2020, as shown in Table 5, significant improvements are achieved on 23 functions. From Tables 6 to 9, we know that the number of functions with "+" decreases compared with the results shown in Tables 2–5. Meanwhile, from Tables 6 to 9, only a few functions show performance degradation after applying the ESM. We can therefore conclude that the ESM is still competitive even in recognized excellent DE variants.

We further conducted Wilcoxon signed-ranked test with a 0.05 significance [40] to statistically verify the effectiveness of the ESM. In Table 10, the p-value obtained by ESM-DE/rand/1, ESM-DE/best/1, ESM-DE/current − to − best/1, ESM-j2020, and ESM-BeSD are all less than 0.05, which indicates a significant improvement is achieved by the proposed ESM. It also can be found that ESM-EJADE, ESM-EBSHADE, and ESM-EBLSHADE still maintain their advantages in terms of the value of $R+$ and R−, which also reflects that the ESM achieves more promising results.

As mentioned before, the functions of CEC2017 can be divided into four categories: unimodal functions, simple multimodal functions, hybrid functions, and composition functions. To analyze whether the performance of the ESM is related to the type of function, the performance of the ESM-based algorithms for each type of function category are presented in Table 11. As shown in Table 11, the ESM performs best on hybrid functions. Following the adoption of the ESM, the algorithm has an 81.25% probability of finding a better solution but only a 16.25% probability of finding a worse solution on hybrid functions. Besides, the performance of the ESM is almost the same on simple multimodal functions and composition functions and relatively worst on unimodal functions. Table 11 demonstrates that if the optimization problem is multimodal function, hybrid function, or composition function, then the proposed selection mechanism can be considered in the DE algorithm.

Table 1: Parameter settings of all compared algorithms.

No.	ESM	A = NP, Q = 120
1	DE/rand/1	$F = 0.5$, CR = 0.9;
2	DE/best/1	$F = 0.5$, CR = 0.5;
3	DE/current − to − best/1	$F = 0.5$, CR = 0.5;
4	j2020 [37]	$F = 0.5$, CR = 0.9;
5	EJADE [38]	$u_F = 0.5$, $u_{\mathrm{CR}} = 0.5$, $c = 0.1$, $p = 0.05$;
6	BeSD [39]	$K = 0.5$;
7	EBSHADE [14]	$M_F = 0.5$, $M_{\mathrm{CR}} = 0.5$, and $H = 100$;
8	EBLSHADE [14]	$M_F = 0.5$, $M_{\mathrm{CR}} = 0.5$, and $H = 100$;

5.2. Convergence Analysis. To more clearly show the influence of the ESM on algorithm convergence and avoid diversion, four representative functions from CEC2017 are extracted, including F7, F16, F20, and F24, convergence curves for these functions with the three classic DEs and their relative ESM-based versions are presented. F7 is a simple multimodal function with a characteristic of convexity. From Figure 5(a), we can find that all six comparison algorithms have a similar convergence rate; this also reflects that the ESM does not cripple the convergence speed of the algorithm on F7. From Figures 5(b) to 5(d), it can be seen that when each algorithm adopts the proposed ESM, it converges to a better value than the original algorithm with faster convergence. This is mainly because the algorithm can hardly escape when it suffers stagnation. The algorithm that uses the proposed ESM has a more excellent capability to keep search efficiency.

5.3. Population Diversity Analysis. As the ESM adopts the entropy-based individual diversity measure, we further calculated the diversity of the main population and archive population on F7, F16, F20, and F24 for three classic DEs. It should be pointed out that we used the concept proposed in [41] to measure population diversity:

$$\begin{aligned}\overline{x_j} &= \frac{1}{\mathrm{NP}}\sum_{i=1}^{\mathrm{NP}} x_{j,i},\\ \text{Population Diversity} &= \sqrt{\frac{1}{\mathrm{NP}}\sum_{i=1}^{\mathrm{NP}}\sum_{j=1}^{D}\left(x_{j,i} - \overline{x_j}\right)^2}.\end{aligned} \tag{13}$$

As illustrated in Figure 6(a), the main population and archive population have an almost uniform diversity on F7. From Figures 6(b)to6(d), it can be concluded that the archive population can maintain a relatively higher diversity than the main population at the intermediate stage for DE/current − to − best/1 and DE/best/1. To sum up, the archive population can maintain relative diversity during the search process, which makes it possible for the algorithm to recover from stagnation.

Table 2: Comparison results of DE/rand/1 on CEC2017 test suite $(D = 30)$.

	DE/rand/1		ESM-DE/rand/1		
	MEan	Std.DEv	MEan	Std.DEv	
F1	5.8903E − 01	2.9710E − 01	**4.5086E − 01**	1.7552E − 01	+
F2	5.7444E + 23	9.9418E + 23	**9.2675E + 21**	2.5025E + 22	+
F3	3.4902E + 04	6.1469E + 03	**2.3752E + 04**	3.9591E + 03	+
F4	**6.5749E + 01**	9.5494E + 00	6.7267E + 01	1.0601E + 01	−
F5	1.7859E + 02	9.8332E + 00	**1.7115E + 02**	1.1998E + 01	+
F6	1.3560E − 06	1.8981E − 06	6.2629E − 07	4.5960E − 07	=
F7	2.1873E + 02	1.0943E + 01	**2.1104E + 02**	8.6729E + 00	+
F8	1.8310E + 02	9.0762E + 00	**1.7903E + 02**	7.5831E + 00	+
F9	0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	=
F10	6.8115E + 03	2.7719E + 02	**6.7908E + 03**	2.7506E + 02	+
F11	7.7115E + 01	1.8139E + 01	**7.6014E + 01**	2.7738E + 01	+
F12	1.9220E + 06	1.3261E + 06	**1.3328E + 05**	1.6493E + 05	+
F13	6.9480E + 02	1.3966E + 02	**4.9388E + 02**	8.5027E + 01	+
F14	7.6534E + 01	6.7506E + 00	**7.3477E + 01**	6.1189E + 00	+
F15	7.2091E + 01	6.3029E + 00	**6.0288E + 01**	7.8284E + 00	+
F16	1.1650E + 03	1.6343E + 02	**9.3936E + 02**	1.6736E + 02	+
F17	2.8488E + 02	6.0664E + 01	**7.4398E + 01**	8.8628E + 00	+
F18	1.3888E + 04	5.2100E + 03	**8.0349E + 03**	2.2066E + 03	+
F19	3.6625E + 01	3.4612E + 00	**3.6208E + 01**	3.0908E + 00	+
F20	1.9409E + 02	1.4131E + 02	**3.4660E + 01**	6.7089E + 00	+
F21	3.7093E + 02	9.2962E + 00	**3.6776E + 02**	1.0301E + 01	+
F22	1.0000E + 02	1.5498E − 13	1.0000E + 02	0.0000E + 00	=
F23	5.2446E + 02	9.2278E + 00	**5.1863E + 02**	8.4044E + 00	+
F24	5.8819E + 02	1.2276E + 01	**5.8615E + 02**	8.0042E + 00	+
F25	3.8675E + 02	2.5661E − 02	3.8674E + 02	1.1873E − 02	=
F26	2.6453E + 03	9.4563E + 01	**2.5338E + 03**	1.1013E + 02	+
F27	**4.9203E + 02**	1.0543E + 01	4.9375E + 02	8.0730E + 00	−
F28	3.2756E + 02	4.6815E + 01	**3.1464E + 02**	3.5907E + 01	+
F29	9.0769E + 02	9.6800E + 01	**7.3123E + 02**	7.3738E + 01	+
F30	1.5880E + 04	3.3864E + 03	**1.0375E + 04**	1.4700E + 03	+
Total numbEr of (+/=/−): 24/4/2					

Table 3: Comparison results of DE/best/1 on CEC2017 test suite $(D = 30)$.

	DE/best/1		ESM-DE/best/1		
	Mean	Std.Dev	Mean	Std.Dev	
F1	0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	=
F2	2.8258E + 10	1.1400E + 11	**3.5454E + 04**	1.6884E + 05	+
F3	6.0493E + 02	3.5493E + 02	**3.1110E + 00**	4.8214E + 00	+
F4	9.1949E + 01	2.7991E + 01	9.2586E + 01	2.5015E + 01	=
F5	9.2763E + 01	5.1334E + 01	**4.4342E + 01**	1.6997E + 01	+
F6	2.5865E − 02	6.1852E − 02	3.6252E − 02	1.1222E − 01	=
F7	1.7540E + 02	3.2845E + 01	**8.2894E + 01**	2.9691E + 01	+
F8	9.1332E + 01	5.5536E + 01	**4.5522E + 01**	1.1158E + 01	+
F9	1.3824E + 01	2.8355E + 01	1.3216E + 01	2.0467E + 01	=
F10	6.1379E + 03	5.6240E + 02	**2.2535E + 03**	1.6207E + 03	+
F11	6.9152E + 01	4.1818E + 01	**5.9835E + 01**	3.3934E + 01	+
F12	2.4650E + 04	1.7178E + 04	**1.8722E + 04**	1.1910E + 04	+
F13	**8.0829E + 03**	1.3281E + 04	1.1159E + 04	1.4904E + 04	−
F14	1.0157E + 02	2.4475E + 01	**7.9954E + 01**	3.5773E + 01	+
F15	1.3412E + 02	7.9017E + 01	**8.4741E + 01**	5.4255E + 01	+
F16	**4.6549E + 02**	2.8862E + 02	4.8784E + 02	2.7012E + 02	−
F17	1.6363E + 02	9.3907E + 01	**1.5320E + 02**	8.2527E + 01	+
F18	7.5687E + 04	5.7404E + 04	**4.6144E + 04**	2.9812E + 04	+
F19	6.4986E + 01	4.8051E + 01	**6.0751E + 01**	5.8037E + 01	+
F20	1.6043E + 02	9.2239E + 01	1.6611E + 02	9.9222E + 01	=
F21	2.8238E + 02	4.5823E + 01	**2.4758E + 02**	1.1434E + 01	+
F22	2.8353E + 03	2.8108E + 03	**1.2129E + 03**	1.6013E + 03	+

Table 3: Continued.

	DE/best/1		ESM-DE/best/1		
	Mean	Std.Dev	Mean	Std.Dev	
F23	4.1270E + 02	2.4531E + 01	**3.9811E + 02**	1.4860E + 01	+
F24	**4.7466E + 02**	2.8842E + 01	4.7812E + 02	1.3574E + 01	−
F25	3.9123E + 02	1.1628E + 01	3.9129E + 02	1.1172E + 01	=
F26	1.6222E + 03	2.1578E + 02	**1.4146E + 03**	4.2851E + 02	+
F27	**5.1391E + 02**	1.5952E + 01	5.1741E + 02	1.2805E + 01	−
F28	4.3979E + 02	2.9759E + 01	**4.2865E + 02**	2.3995E + 01	+
F29	**5.7692E + 02**	1.1599E + 02	5.9473E + 02	1.0825E + 02	−
F30	6.3732E + 03	3.1823E + 03	**4.2434E + 03**	2.5536E + 03	+
Total number of (+/=/−): 19/6/5					

Table 4: Comparison results of DE/current − to − best/1 on CEC2017 test suite ($D = 30$).

	DE/current − to − best/1		ESM-DE/current − to − best/1		
	Mean	Std.Dev	Mean	Std.Dev	
F1	4.2529E + 03	1.3861E + 04	**2.4194E + 03**	4.8764E + 03	+
F2	**1.4541E + 13**	5.2565E + 13	7.0969E + 13	3.8743E + 14	−
F3	2.5651E + 02	1.7770E + 02	**1.4208E + 00**	2.2041E + 00	+
F4	1.1469E + 02	1.3856E + 01	**1.0468E + 02**	1.8355E + 01	+
F5	1.4448E + 02	1.2098E + 01	**2.9207E + 01**	6.2955E + 00	+
F6	7.9340E − 03	4.4121E − 02	**2.0523E − 03**	1.1284E − 02	+
F7	1.7631E + 02	1.2016E + 01	**7.5749E + 01**	3.9251E + 01	+
F8	1.4187E + 02	1.2879E + 01	**2.7608E + 01**	7.8113E + 00	+
F9	4.2194E − 01	4.5290E − 01	**2.8916E − 01**	3.5412E − 01	+
F10	6.1656E + 03	3.9958E + 02	**4.6688E + 03**	1.8249E + 03	+
F11	6.8524E + 01	3.1083E + 01	**5.5608E + 01**	2.9379E + 01	+
F12	2.2269E + 04	1.3508E + 04	**1.9785E + 04**	1.3868E + 04	+
F13	6.5569E + 03	3.5373E + 03	**4.7370E + 03**	7.3061E + 03	+
F14	1.2361E + 02	2.9917E + 01	**8.7387E + 01**	2.5164E + 01	+
F15	2.1775E + 02	8.1693E + 01	**1.9212E + 02**	8.3283E + 01	+
F16	7.4089E + 02	2.3808E + 02	**4.1651E + 02**	2.2185E + 02	+
F17	1.8434E + 02	5.3907E + 01	**9.5323E + 01**	5.6810E + 01	+
F18	5.8340E + 04	2.6259E + 04	**5.2731E + 04**	3.7186E + 04	+
F19	**7.9602E + 01**	2.9727E + 01	1.1554E + 02	5.1436E + 01	−
F20	2.2812E + 02	8.2020E + 01	**1.2592E + 02**	8.0240E + 01	+
F21	3.3232E + 02	1.0378E + 01	**2.3302E + 02**	1.8782E + 01	+
F22	2.9659E + 02	1.0908E + 03	**1.0053E + 02**	1.1111E + 00	+
F23	4.6069E + 02	2.1824E + 01	**3.8629E + 02**	1.4205E + 01	+
F24	5.4998E + 02	1.6167E + 01	**4.6028E + 02**	1.5954E + 01	+
F25	4.0471E + 02	1.7220E + 01	4.0175E + 02	1.7238E + 01	=
F26	1.9292E + 03	5.1027E + 02	**1.3090E + 03**	2.7680E + 02	+
F27	5.1686E + 02	1.1760E + 01	5.1961E + 02	1.4919E + 01	=
F28	4.7064E + 02	3.4376E + 01	**4.5744E + 02**	3.0111E + 01	+
F29	7.0816E + 02	1.0684E + 02	**5.2205E + 02**	8.3692E + 01	+
F30	9.8358E + 03	7.4280E + 03	**7.5623E + 03**	3.6244E + 03	+
Total number of (+/=/−): 26/2/2					

5.4. Parameter Sensitivity Analysis. As illustrated in the previous sections, the ESM contains one parameter Q required adjustment. Q is a threshold which represents the stagnation steps. Table 12 shows the B-W values that represent the variation between better ("+") and worse ("−") number of the mean value after applying the ESM. It can be seen that the ESM-based algorithm performance is not so sensitive to the change of Q values over a finite range. For further comparison, A wider range of Q values and its corresponding B-W values are shown in Figure 7. In general, when the Q value is too large, the overall performance of the ESM begins to deteriorate. This is mainly because too large Q value reduces the probability of the ESM intervening in the algorithm; therefore, it is difficult for the ESM to function in limited iterations. In summary, considering the overall performance of the ESM, Q is set to 120.

5.5. Scalability Analysis. The dimension of the test functions governs the difficulties on finding the global optimum. Higher dimensional functions are generally more difficult to

Table 5: Comparison results of j2020 on CEC2017 test suite (D = 30).

	j2020		ESM-j2020		
	Mean	Std.Dev	Mean	Std.Dev	
F1	**3.9138E − 02**	2.7831E − 02	4.1149E − 02	2.6801E − 02	−
F2	2.1525E + 22	7.2306E + 22	**3.5391E + 16**	7.4152E + 16	+
F3	3.0526E + 04	5.6182E + 03	**1.5964E + 04**	3.6765E + 03	+
F4	**7.6190E + 01**	1.0125E + 01	7.7537E + 01	1.0041E + 01	−
F5	1.5059E + 02	1.5882E + 01	**3.3617E + 01**	1.1325E + 01	+
F6	0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	=
F7	1.9215E + 02	1.0204E + 01	**8.8048E + 01**	2.4532E + 01	+
F8	1.5771E + 02	9.7337E + 00	**3.5822E + 01**	1.3602E + 01	+
F9	0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	=
F10	6.0724E + 03	2.8112E + 02	**2.6253E + 03**	9.8373E + 02	+
F11	7.1620E + 01	2.5123E + 01	**1.4904E + 01**	1.6548E + 01	+
F12	7.3213E + 05	6.1498E + 05	**4.9652E + 04**	3.2587E + 04	+
F13	3.9600E + 02	8.0250E + 01	**2.6856E + 02**	4.2577E + 01	+
F14	7.1081E + 01	5.4527E + 00	**5.2870E + 01**	1.1383E + 01	+
F15	6.0349E + 01	5.8258E + 00	**3.6537E + 01**	1.3383E + 01	+
F16	9.0601E + 02	1.4229E + 02	**4.1188E + 02**	1.7961E + 02	+
F17	9.1833E + 01	3.4540E + 01	**4.3249E + 01**	2.4777E + 01	+
F18	3.2864E + 03	1.5646E + 03	**1.5206E + 03**	8.9776E + 02	+
F19	3.3522E + 01	4.8067E + 00	**1.5961E + 01**	2.9754E + 00	+
F20	5.3644E + 01	2.1292E + 01	**4.7332E + 01**	6.8896E + 01	+
F21	3.5434E + 02	6.7332E + 00	**2.3149E + 02**	1.2889E + 01	+
F22	1.0000E + 02	2.2819E − 13	1.0000E + 02	8.1676E − 14	=
F23	4.9590E + 02	6.7701E + 00	**3.7594E + 02**	1.0453E + 01	+
F24	5.7509E + 02	8.0847E + 00	**4.5326E + 02**	1.1284E + 01	+
F25	3.8677E + 02	2.5429E − 02	3.8676E + 02	1.3848E − 02	=
F26	2.3688E + 03	1.1160E + 02	**1.1769E + 03**	1.2856E + 02	+
F27	**4.9473E + 02**	1.1847E + 01	4.9543E + 02	6.3871E + 00	−
F28	3.3192E + 02	4.5936E + 01	**3.2368E + 02**	4.8265E + 01	+
F29	7.8452E + 02	8.7484E + 01	**4.8871E + 02**	2.6830E + 01	+
F30	9.8924E + 03	2.2550E + 03	**6.4595E + 03**	1.2161E + 03	+
Total numbEr of (+/=/−): 23/4/3					

Table 6: Comparison results of EJADE on CEC2017 test suite (D = 30).

	EJADE		ESM-EJADE		
	Mean	Std.Dev	Mean	Std.Dev	
F1	0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	=
F2	**4.5316E + 03**	2.1794E + 04	1.2673E + 04	5.6411E + 04	−
F3	0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	=
F4	**2.4107E + 01**	2.9037E + 01	2.6740E + 01	2.9430E + 01	−
F5	1.8962E + 01	4.4630E + 00	**1.7560E + 01**	4.1722E + 00	+
F6	**6.9845E − 07**	2.4275E − 06	1.4467E − 06	6.4530E − 06	−
F7	4.6647E + 01	2.8902E + 00	**4.5802E + 01**	3.2092E + 00	+
F8	1.8850E + 01	3.1904E + 00	**1.7218E + 01**	5.1087E + 00	+
F9	1.9587E − 01	3.4505E − 01	**1.2548E − 01**	1.8845E − 01	+
F10	1.8437E + 03	3.0553E + 02	**1.5558E + 03**	5.3726E + 02	+
F11	4.7182E + 01	2.6517E + 01	**3.9138E + 01**	2.3463E + 01	+
F12	2.0542E + 03	1.8865E + 03	**1.8759E + 03**	1.2938E + 03	+
F13	4.9648E + 01	3.1191E + 01	**3.7663E + 01**	1.7336E + 01	+
F14	3.9862E + 01	1.3835E + 01	**3.8508E + 01**	1.1120E + 01	+
F15	**4.1678E + 01**	4.0043E + 01	4.2540E + 01	3.6838E + 01	−
F16	3.2052E + 02	1.8067E + 02	**2.6554E + 02**	1.5914E + 02	+
F17	5.7500E + 01	1.3465E + 01	**5.1900E + 01**	3.5779E + 01	+
F18	**7.7806E + 01**	5.1375E + 01	8.8347E + 01	6.3700E + 01	−
F19	**1.7808E + 01**	1.0382E + 01	1.9606E + 01	1.8146 E + 01	−
F20	1.0632E + 02	5.7664E + 01	**5.1910E + 01**	5.3314E + 01	+
F21	2.1850E + 02	3.6330E + 00	**2.1748E + 02**	4.0920E + 00	+

Table 6: Continued.

	EJADE		ESM-EJADE		
	Mean	Std.Dev	Mean	Std.Dev	
F22	**1.0008E + 02**	4.4179E − 01	1.6606E + 02	3.6781E + 02	−
F23	3.6253E + 02	6.0195E + 00	**3.6162E + 02**	6.1887E + 00	+
F24	4.3448E + 02	6.1456E + 00	4.3460E + 02	5.0372E + 00	=
F25	3.8703E + 02	7.8427E − 01	3.8701E + 02	3.6023E − 01	=
F26	1.1198E + 03	7.2825E + 01	**1.0310E + 03**	2.7881E + 02	+
F27	5.0690E + 02	8.8147E + 00	5.0681E + 02	6.6749E + 00	=
F28	3.7620E + 02	6.0263E + 01	**3.3258E + 02**	5.2414E + 01	+
F29	4.6359E + 02	2.7079E + 01	**4.4343E + 02**	3.0038E + 01	+
F30	2.1586E + 03	1.9675E + 02	**2.1400E + 03**	2.0292E + 02	+
Total number of (+/ = /−): 18/5/7					

Table 7: Comparison results of BeSD on CEC2017 test suite $(D = 30)$.

	BeSD		ESM- BeSD		
	Mean	Std.Dev	Mean	Std.Dev	
F1	0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	=
F2	7.2258E + 00	1.9085E + 01	**3.9032E + 00**	1.4996E + 01	+
F3	0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	=
F4	**6.8185E + 00**	1.6230E + 01	1.7131E + 01	2.6599E + 01	−
F5	4.5762E + 01	2.8543E + 01	**2.7217E + 01**	8.1867E + 00	+
F6	0.0000E + 00	0.0000E + 00	1.7219E − 07	3.7418E − 07	=
F7	1.3363E + 02	1.5180E + 01	**5.1262E + 01**	6.7117E + 00	+
F8	5.6343E + 01	2.9912E + 01	**2.6864E + 01**	7.8132E + 00	+
F9	6.1295E − 02	1.9377E − 01	6.7718E − 02	1.7464E − 01	=
F10	5.8972E + 03	3.7315E + 02	**2.5464E + 03**	5.6311E + 02	+
F11	**2.6612E + 01**	1.5396E + 01	3.7557E + 01	2.2784E + 01	−
F12	**8.3133E + 03**	4.3141E + 03	8.9538E + 03	7.0576E + 03	−
F13	2.2656E + 02	2.3629E + 02	**6.3469E + 01**	2.8828E + 01	+
F14	6.1777E + 01	1.5857E + 01	**4.1127E + 01**	9.6938E + 00	+
F15	6.2060E + 01	3.5382E + 01	**3.8429E + 01**	2.6456E + 01	+
F16	6.1366E + 02	1.7387E + 02	**2.2941E + 02**	1.8203E + 02	+
F17	1.0414E + 02	2.0838E + 01	**3.4358E + 01**	1.1387E + 01	+
F18	2.1008E + 02	9.9628E + 01	**7.7754E + 01**	4.7271E + 01	+
F19	4.0368E + 01	2.1662E + 01	**3.0129E + 01**	1.4941E + 01	+
F20	1.1522E + 02	4.3894E + 01	**4.3747E + 01**	5.1852E + 01	+
F21	2.3501E + 02	2.5203E + 01	**2.2421E + 02**	9.6408E + 00	+
F22	1.0000E + 02	1.1357E − 13	1.0000E + 02	2.2124E − 13	=
F23	3.7384E + 02	1.6564E + 01	3.7369E + 02	9.3329E + 00	=
F24	4.4032E + 02	7.1928E + 00	**4.3973E + 02**	8.2282E + 00	+
F25	3.8708E + 02	2.9529E − 01	3.8709E + 02	2.7823E − 01	=
F26	1.1447E + 03	1.7473E + 02	**1.0790E + 03**	2.7897E + 02	+
F27	**5.0192E + 02**	5.4148E + 00	5.0552E + 02	8.0174E + 00	−
F28	**3.1676E + 02**	3.8969E + 01	3.3033E + 02	4.8241E + 01	−
F29	5.9423E + 02	7.0224E + 01	**4.4240E + 02**	3.4337E + 01	+
F30	2.5624E + 03	3.5089E + 02	**2.1699E + 03**	1.3768E + 02	+
Total number of (+/ = /−): 18/7/5					

solve. To verify the relationship between the dimensionality and the performance of the proposed mechanism, we evaluate the average performance of eight ESM-based DE algorithms in CEC2017 benchmark set. Table 13 shows the results (+/=/−) of the considered algorithms at $D = 10$, 30, 50, and 100 on the test functions, for intuition, we convert the corresponding B-W values into Figure 8. We can find that the performance of the ESM-based algorithm fluctuates slightly but does not degrade significantly with the increase of problem dimension compared to the original algorithm. In Table 14, the Friedman test [40], a widely used non-parametric test in the EA community, is used to validate the performance of all algorithms based on the mean value. It is not difficult to see that the p-values from $D = 10$ to $D = 100$ calculated by the Friedman test are all less than 0.05. Therefore, there are significant differences in the performance of the comparison algorithms on the corresponding dimensions, and ESM-EBLSHADE gets the first rank overall.

Table 8: Comparison results of EBSHADE on CEC2017 test suite ($D = 30$).

	EBSHADE		ESM-EBSHADE		
	Mean	Std.Dev	Mean	Std.Dev	
F1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	=
F2	1.9972E+04	1.1037E+05	**1.6852E+02**	7.3798E+02	+
F3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	=
F4	5.6852E+01	1.0598E+01	**5.3253E+01**	1.7773E+01	+
F5	**1.5508E+01**	2.5802E+00	1.6273E+01	6.3554E+00	−
F6	**6.8453E−08**	1.0785E-07	4.2769E −07	7.4531E −07	−
F7	4.6604E+01	2.7607E+00	**4.6489E+01**	4.8631E+00	+
F8	1.6879E+01	2.8304E+00	**1.6186E+01**	3.4825E+00	+
F9	1.7544E−02	8.2641E −02	**1.4656E−02**	8.1599E −02	+
F10	1.6353E+03	2.3269E+02	**1.4154E+03**	2.6504E+02	+
F11	2.7724E+01	2.6071E+01	**2.6265E+01**	2.5887E+01	+
F12	1.2219E+03	4.6754E+02	**1.2126E+03**	4.4003E+02	+
F13	3.8215E+01	2.0185E+01	**3.7691E+01**	1.4370E+01	+
F14	**2.8375E+01**	8.7055E+00	3.0017E+01	4.8689E+00	−
F15	**1.7726E+01**	1.0888E+01	2.1258E+01	1.2917E+01	−
F16	2.9276E+02	1.1820E+02	**2.0707E+02**	1.2958E+02	+
F17	4.5658E+01	1.0476E+01	3.5719E+01	1.0310E+01	=
F18	**9.0145E+01**	4.7859E+01	9.4472E+01	5.6305E+01	−
F19	1.4666E+01	8.9947E+00	**1.2179E+01**	4.1749E+00	+
F20	6.3748E+01	4.4406E+01	**5.2312E+01**	4.9270E+01	+
F21	**2.1777E+02**	3.4870E+00	2.1847E+02	4.1234E+00	−
F22	1.0051E+02	2.8400E+00	1.0000E+02	2.2124E −13	=
F23	3.6665E+02	5.5240E+00	**3.6453E+02**	5.2131E+00	+
F24	4.3676E+02	3.7374E+00	4.3642E+02	6.0049E+00	=
F25	3.8681E+02	5.8187E-02	3.8684E+02	7.5494E −02	=
F26	1.1214E+03	6.7789E+01	**1.1160E+03**	7.3256E+01	+
F27	5.0632E+02	4.6020E+00	**5.0579E+02**	6.3243E+00	+
F28	3.4497E+02	5.8589E+01	**3.4205E+02**	5.8955E+01	+
F29	4.6900E+02	1.3609E+01	**4.4219E+02**	2.2017E+01	+
F30	2.0864E+03	1.3828E+02	**2.0213E+03**	1.6473E+02	+
Total number of (+/=/−): 18/6/6					

Table 9: Comparison results of EBLSHADE on CEC2017 test suite ($D = 30$).

	EBLSHADE		ESM-EBLSHADE		
	Mean	Std.Dev	Mean	Std.Dev	
F1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	=
F2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	=
F3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	=
F4	5.8562E+01	3.4396E−14	5.8562E+01	4.0776E−14	=
F5	6.6978E+00	1.3325E+00	6.9719E+00	1.5418E+00	=
F6	0.0000E+00	0.0000E+00	1.4398E−08	4.1347E−08	=
F7	3.7938E+01	1.3728E+00	**3.6957E+01**	1.2423E+00	+
F8	7.4627E+00	1.5388E+00	**6.9654E+00**	1.6395E+00	+
F9	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	=
F10	1.3903E+03	2.5407E+02	**1.2476E+03**	2.3487E+02	+
F11	**3.3748E+01**	2.8992E+01	4.0898E+01	2.8428E+01	−
F12	1.1107E+03	3.2096E+02	**1.1021E+03**	3.2418E+02	+
F13	1.8932E+01	7.7131E+00	**1.6355E+01**	5.4803E+00	+
F14	2.1636E+01	1.1516E+00	**2.1493E+01**	3.6187E+00	+
F15	3.9785E+00	1.8483E+00	**3.6417E+00**	1.6998E+00	+
F16	4.1583E+01	3.4857E+01	**2.0657E+01**	5.3775E+00	+
F17	3.1880E+01	5.6772E+00	**3.1060E+01**	4.8829E+00	+
F18	**2.2262E+01**	1.5977E+00	2.2496E+01	1.2879E+00	−
F19	6.2724E+00	2.2466E+00	**5.6099E+00**	1.3474E+00	+
F20	3.2672E+01	6.5190E+00	**2.7681E+01**	6.2855E+00	+
F21	2.0835E+02	1.4350E+00	**2.0819E+02**	1.4361E+00	+
F22	1.0000E+02	0.0000E+00	1.0000E+02	8.1676E−14	=

Table 9: Continued.

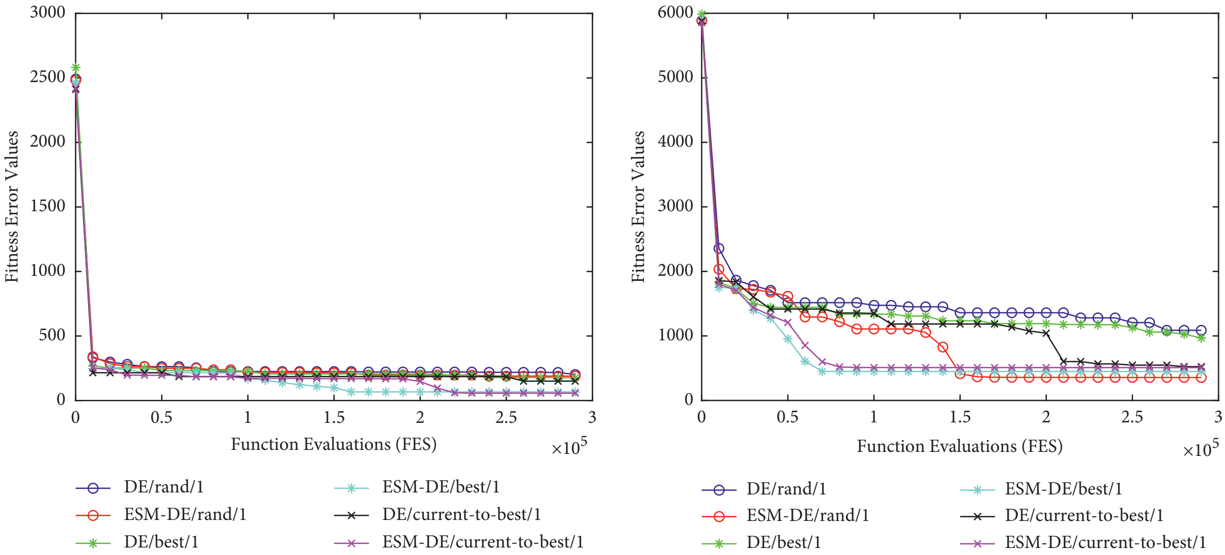
	EBLSHADE		ESM-EBLSHADE		
	Mean	Std.Dev	Mean	Std.Dev	
F23	**3.5311E + 02**	3.0584*E* + 00	3.5407*E* + 02	2.9996*E* + 00	−
F24	**4.2796E + 02**	1.7733*E* + 00	4.2807*E* + 02	1.8546*E* + 00	−
F25	3.8672*E* + 02	1.9899*E* − 02	3.8672*E* + 02	2.4427*E* − 02	=
F26	9.7950*E* + 02	4.0916*E* + 01	**9.7385E + 02**	2.9174*E* + 01	+
F27	5.0689*E* + 02	5.5012*E* + 00	**5.0580E + 02**	4.6504*E* + 00	+
F28	3.5816*E* + 02	6.2765*E* + 01	**3.2838E + 02**	4.8986*E* + 01	+
F29	4.3621*E* + 02	7.9105*E* + 00	**4.3392E + 02**	6.5290*E* + 00	+
F30	**1.9967E + 03**	7.3147*E* + 01	2.0385*E* + 03	8.6201*E* + 01	−
Total number of (+/=/−): 16/9/5					

Table 10: Results obtained by the Wilcoxon test.

Algorithms	*R*+	*R*−	*P*-value
ESM-DE/rand/1 vs. DE/rand/1	**446.5**	18.5	**0.039**
ESM-DE/best/1 vs. DE/best/1	**376.5**	88.5	**0.047**
ESM-DE/current − to − best/1 vs. DE/current − to − best/1	**423**	42	**0.031**
ESM-j2020 vs. j2020	**419.5**	45.5	**0.038**
ESM-EJADE vs. EJADE	**326.5**	138.5	0.081
ESM-BeSD vs. BeSD	**338.5**	126.5	**0.046**
ESM-EBSHADE vs. EBSHADE	**375.5**	89.5	0.075
ESM-EBLSHADE vs. EBLSHADE	**341.5**	123.5	0.089

Table 11: The classification statistics result of proposed ESM in eight algorithms.

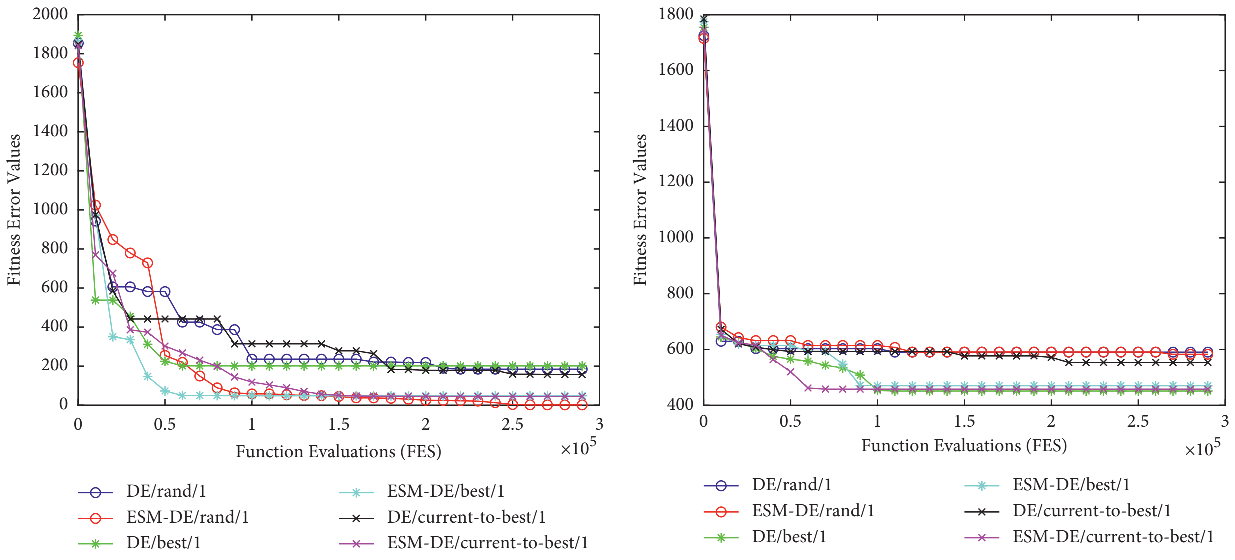
Functions	+(%)	=(%)	−(%)
Unimodal functions	45.83	41.67	12.50
Simple multimodal functions	64.29	23.21	12.50
Hybrid functions	**81.25**	2.50	16.25
Composition functions	62.50	22.50	15.00



(a)

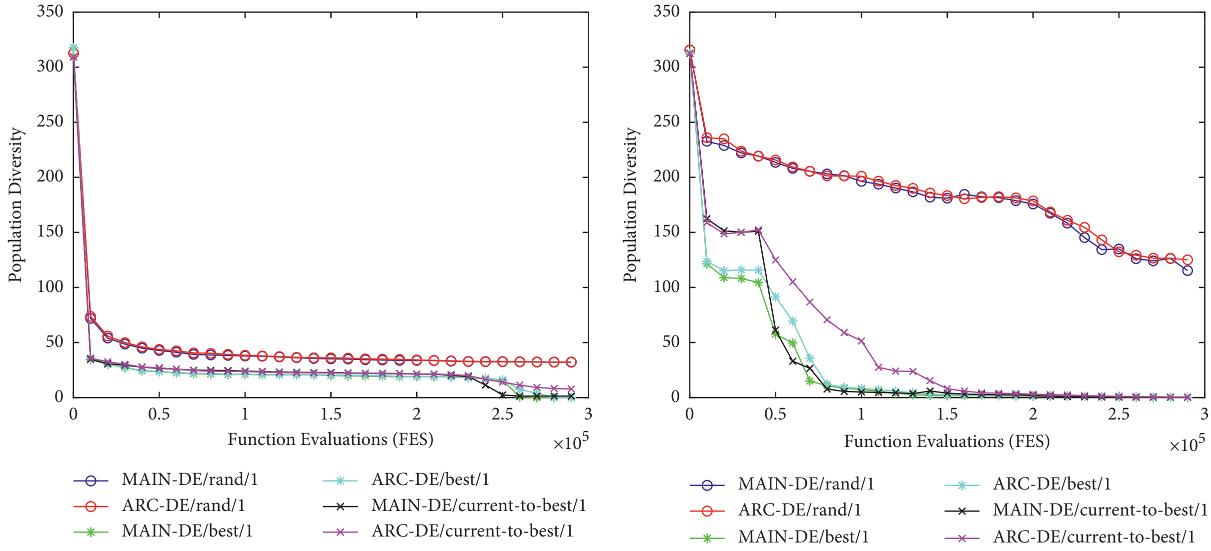
(b)

Figure 5: Continued.



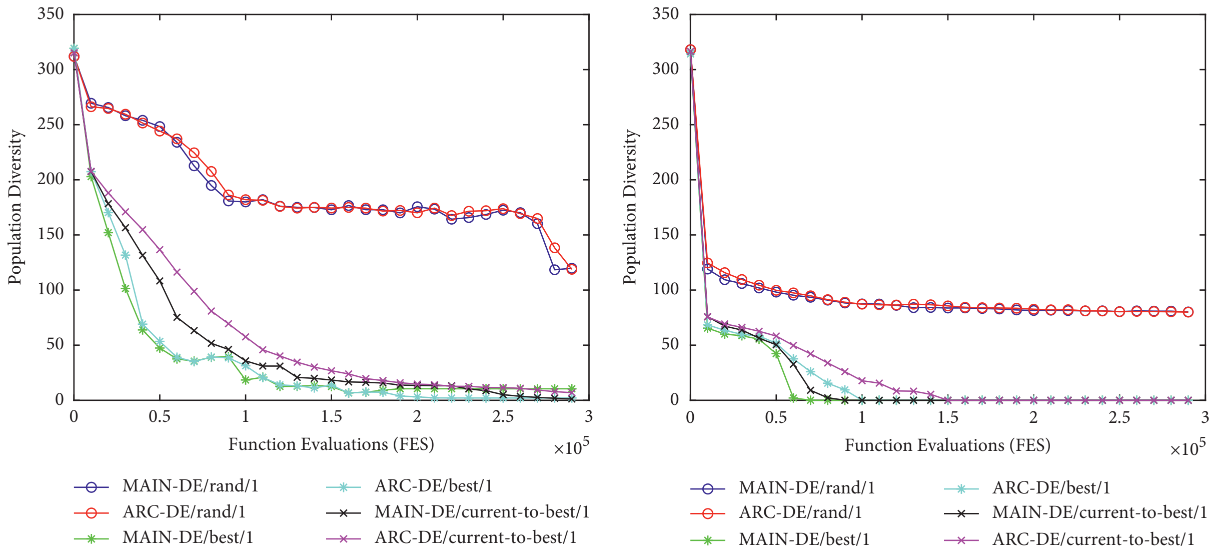
(c) (d)

Figure 5: Convergence curves of three classic DEs and relative ESM-DEs on four 30-D CEC2017 benchmark functions: (a) F7; (b) F16; (c) F20; (d) F24.



(a) (b)

Figure 6: Continued.



(c)

(d)

Figure 6: Population diversity curves of main population and archive population of three classic DEs on four 30-D CEC2017 benchmark functions: (a) F7; (b) F16; (c) F20; (d) F24.

Table 12: The B-W values of all ESM-based algorithms on CEC2017 under different Q values.

Algorithms	Q = 100	Q = 110	Q = 120	Q = 130	Q = 140
ESM-DE/rand/1	22	21	22	23	20
ESM-DE/best/1	13	14	14	13	14
ESM-DE/current – to – best/1	24	24	24	22	20
ESM-j2020	18	21	20	19	17
ESM-EJADE	10	12	11	10	10
ESM-BeSD	11	10	13	12	12
ESM-EBSHADE	10	11	12	10	11
ESM-EBLSHADE	9	10	11	10	10
Total	117	126	**127**	119	114

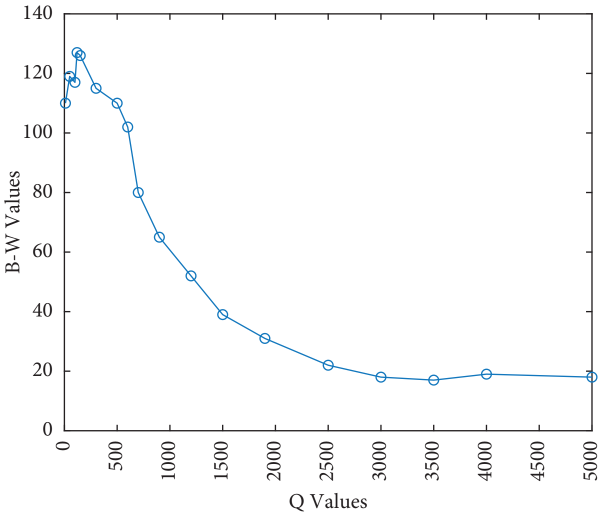


Figure 7: The B-W values at different Q values.

Table 13: The statistical results of all ESM-based algorithms on CEC2017 under different dimensions.

Algorithms	D = 10	D = 30	D = 50	D = 100
ESM-DE/rand/1 vs. DE/rand/1	23/5/2	24/4/2	22/6/2	25/4/1
ESM-DE/best/1 vs. DE/best/1	17/8/5	19/6/5	18/6/6	19/5/6
ESM-DE/current − to − best/1 vs. DE/current − to − best/1	23/4/3	26/2/2	25/2/3	23/5/2
ESM-j2020 vs. j2020	22/5/3	23/4/3	21/5/4	22/4/4
ESM-EJADE vs. EJADE	16/8/6	18/5/7	20/5/5	19/5/6
ESM-BeSD vs. BeSD	17/7/6	18/7/5	19/6/5	18/5/7
ESM-EBSHADE vs. EBSHADE	16/9/5	18/6/6	18/5/7	17/6/7
ESM-EBLSHADE vs. EBLSHADE	15/10/5	16/9/5	15/11/4	16/8/6

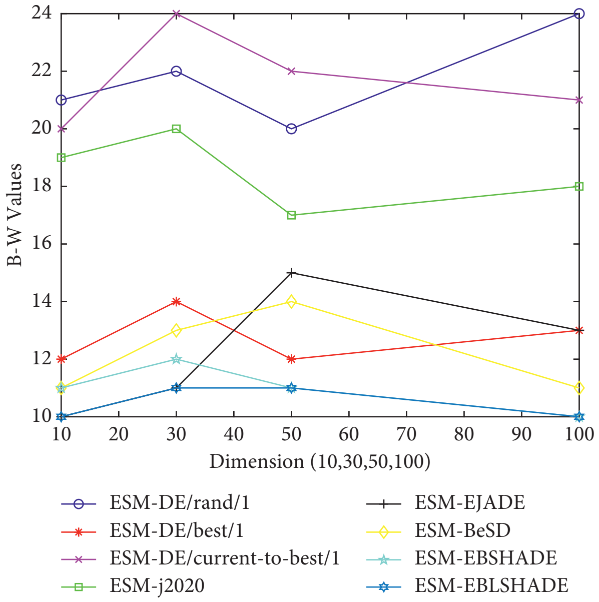


Figure 8: The curves of B-W values for the ESM-based algorithm compared to the original algorithm.

Table 14: The Friedman test results of all algorithms on CEC2017 under different dimensions.

Algorithms	D = 10	D = 30	D = 50	D = 100	Mean ranking	Rank
ESM-EBLSHADE	**2.13**	**2.62**	**2.88**	**2.57**	**10.20**	**1**
EBLSHADE	2.98	3.03	3.21	3.55	12.77	2
ESM-EBSHADE	4.85	4.78	5.01	4.89	19.53	3
EBSHADE	4.98	5.73	5.55	5.21	21.47	4
ESM-EJADE	5.89	6.07	6.24	5.87	24.07	5
ESM-BeSD	5.31	6.73	6.84	6.57	25.45	6
EJADE	6.51	6.93	6.95	6.88	27.27	7
ESM-j2020	7.51	7.62	7.58	7.41	30.12	8
BeSD	7.11	7.93	8.02	8.57	31.63	9
j2020	9.82	10.89	9.58	11.21	41.50	10
ESM-DE/rand/1	10.09	11.03	11.59	11.07	43.78	11
ESM-DE/current − to − best/1	10.55	11.63	11.21	11.85	45.24	12
ESM-DE/best/1	10.11	11.82	11.85	11.98	45.76	13
DE/best/1	11.53	12.65	12.33	12.57	49.08	14
DE/rand/1	12.31	12.70	12.58	12.34	49.93	15
DE/current − to − best/1	11.89	13.83	12.89	12.59	51.20	16
Friedman-P-value	0.00	0.00	0.00	0.00		

6. Conclusion and Future Work

In DE, making full use of the successful solutions generated in iterations is a meaningful work to improve algorithm performance. In this study, an external selection mechanism (ESM) is proposed to restore the searchability of the algorithm when an individual is in a state of stagnation. The ESM mainly contains a successful solution archive mechanism and a crowding entropy diversity control strategy. It can be easily integrated into the existing DE algorithms to improve its performance further.

Experiments are conducted on the CEC2017 benchmark sets cooperated with three classic DEs and five state-of-the-art DE variants. From experiment results, we can see that the ESM-based algorithm can significantly improve the original algorithm's solution accuracy; the ESM also does not increase the computational complexity of the original algorithm due to the introduction of entropy. Further, experiments also show that ESM has a fairly positive effect on multimodal function, hybrid function, and composition function, its scalability on different problem dimensions also has a certain universal.

Notably, experimental results proved that the ESM can effectively improve various DE variants' performance, more study is required. Therefore, our future work will mainly focus on (1) improving the ESM to make it suitable for various single-objective DE; (2) researching the application of the proposed ESM in other evolutionary algorithms.

Data Availability

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

Conflicts of Interest

The authors have no affiliations with or involvement in any organization or entity with any financial interest or nonfinancial interest in the subject matter or materials discussed in this manuscript.

References

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

[2] Q. Liu, S. Du, B. J. van Wyk, and Y. Sun, "Double-layer-clustering differential evolution multimodal optimization by speciation and self-adaptive strategies," *Information Sciences*, vol. 545, no. 1, pp. 465–486, 2021.

[3] A. S. Hameed, B. M. Aboobaider, M. L. Mutar, and N. H. Choon, "A new hybrid approach based on discrete differential evolution algorithm to enhancement solutions of quadratic assignment problem," *International Journal of Industrial Engineering Computations*, vol. 11, no. 1, pp. 51–72, 2020.

[4] A. Bx, A. Hz, B. Mza, and B. Ll, "Differential evolution using cooperative ranking-based mutation operators for constrained optimization," *Swarm and Evolutionary Computation*, vol. 49, pp. 206–219, 2019.

[5] J. Cheng, Z. Pan, H. Liang, Z. Gao, and J. Gao, "Differential evolution algorithm with fitness and diversity ranking-based mutation operator," *Swarm and Evolutionary Computation*, vol. 61, Article ID 100816, 2021.

[6] Z. Beheshti and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithms," *Int. J. Adv. Soft Comput. Appl*, vol. 5, no. 1, pp. 1–35, 2013.

[7] J. Lampinen and I. Zelinka, *On Stagnation of the Differential Evolution Algorithm*, Czech Republic Brno University Of Technology, Brno, Czechia, 2000.

[8] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization," Technical Report, National University of Defense Technology, Changsha, Hunan, 2017.

[9] M. BilalPant, M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, "Differential Evolution: a review of more than two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 90, Article ID 103479, 2020.

[10] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *Journal of Global Optimization*, vol. 27, no. 1, pp. 105–129, 2003.

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

[12] H.-b. Wang, X.-n. Ren, G.-q. Li, and X.-y. Tu, "APDDE: self-adaptive parameter dynamics differential evolution algorithm," *Soft Computing*, vol. 22, no. 4, pp. 1313–1333, 2018.

[13] Y. Zhou, X. Li, and L. Gao, "A differential evolution algorithm with intersect mutation operator," *Applied Soft Computing*, vol. 13, no. 1, pp. 390–401, 2013.

[14] 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.

[15] L. Deng, C. Li, Y. Lan, G. Sun, and C. Shang, "Differential evolution with dynamic combination based mutation operator and two-level parameter adaptation strategy," *Expert Systems with Applications*, vol. 192, Article ID 116298, 2021.

[16] Y. Cai, D. Wu, Y. Zhou, S. Fu, H. Tian, and Y. Du, "Self-organizing neighborhood-based differential evolution for global optimization," *Swarm and Evolutionary Computation*, vol. 56, Article ID 100699, 2020.

[17] L. B. Deng, C. L. Li, and G. J. Sun, "An adaptive dimension level adjustment framework for differential evolution," *Knowledge-Based Systems*, vol. 206, no. 4, Article ID 106388, 2020.

[18] T. J. Choi, J. Togelius, and Y.-G. Cheong, "A fast and efficient stochastic opposition-based learning for differential evolution in numerical optimization," *Swarm and Evolutionary Computation*, vol. 60, Article ID 100768, 2021.

[19] G. Sun, C. Li, and L. Deng, "An adaptive regeneration framework based on search space adjustment for differential evolution," *Neural Computing & Applications*, vol. 33, no. 15, pp. 9503–9519, 2021.

[20] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[21] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE*

Transactions on Evolutionary Computation, vol. 19, no. 4, pp. 508–523, 2014.

[22] L. C. Bezerra, M. López-Ibánez, and T. Stützle, "Automatic component-wise design of multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 403–417, 2015.

[23] Y. Zhou, J. Wang, Y. Zhou, Z. Qiu, Z. Bi, and Y. Cai, "Differential evolution with guiding archive for global numerical optimization," *Applied Soft Computing*, vol. 43, pp. 424–440, 2016.

[24] Z. Zeng, M. Zhang, T. Chen, and Z. Hong, "A new selection operator for differential evolution algorithm," *Knowledge-Based Systems*, vol. 226, Article ID 107150, 2021.

[25] V. Stanovov, S. Akhmedova, and E. Semenkin, "Ranked archive differential evolution with selective pressure for cec 2020 numerical optimization," in *Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, Glasgow, UK, July 2020.

[26] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 4, pp. 623–656, 1948.

[27] Y. G. Petalas, K. E. Parsopoulos, and M. N. Vrahatis, "Entropy-based memetic particle swarm optimization for computing periodic orbits of nonlinear mappings," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, IEEE, Singapore, September 2007.

[28] J. G. Berryman, "Maximum entropy spectral analysis," *Journal of China Institute of Communications*, vol. 17, no. 4, pp. 1519–1533, 2009.

[29] K. M. Malan and A. P. Engelbrecht, "Quantifying ruggedness of continuous landscapes using entropy," in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, Trondheim, Norway, May 2009.

[30] Y. Huang, W. Li, F. Tian, and X. Meng, "A fitness landscape ruggedness multiobjective differential evolution algorithm with a reinforcement learning strategy," *Applied Soft Computing*, vol. 96, no. 1, Article ID 106693, 2020.

[31] L. Lixin Tang, D. Yun Dong, and J. Jiyin Liu, "Differential evolution with an individual-dependent mechanism," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 560–574, 2015.

[32] G. Corriveau, R. Guilbault, A. Tahan, and R. Sabourin, "Review and study of genotypic diversity measures for real-coded representations," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 695–710, 2012.

[33] J. Wang, J. Liao, Y. Zhou, and Y. Cai, "Differential evolution enhanced with multiobjective sorting-based mutation operators," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2792–2805, 2017.

[34] 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.

[35] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: nsga-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[36] N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving cec2017 benchmark problems," in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*, Donostia, Spain, June 2017.

[37] J. Brest, M. S. Maucec, and B. Boskovic, *Differential Evolution Algorithm for Single Objective Bound-Constrained Optimization: Algorithm J2020*, IEEE, Glasgow, UK, 2020.

[38] W. Yi, Y. Chen, Z. Pei, and J. Lu, "Adaptive differential evolution with ensembling operators for continuous optimization problems," *Swarm and Evolutionary Computation*, vol. 69, Article ID 100994, 2021.

[39] P. Civicioglu and E. Besdok, "Bezier search differential evolution algorithm for numerical function optimization," *Expert Systems with Applications*, vol. 165, Article ID 113875, 2021.

[40] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[41] R. Poláková, J. Tvrdík, P. Bujok, and R. Matoušek, *Population-size Adaptation through Diversity-Control Mechanism for Differential Evolution*, IEEE, Vancouver, Canada, 2016.