

Received January 21, 2020, accepted February 18, 2020, date of publication February 20, 2020, date of current version March 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975468

An Estimation of Distribution Algorithm With Multi-Leader Search

XIAOFEI WANG¹, (Member, IEEE), TONG HAN, AND HUI ZHAO

Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China

Corresponding author: Xiaofei Wang (wxf825421673@163.com)

ABSTRACT The estimation of distribution algorithm (EDA) is a well-known stochastic search method but is easily affected by the ill-shaped distribution of solutions and can thus become trapped in stagnation. In this paper, we propose a novel modified EDA with a multi-leader search (MLS) mechanism, namely, the MLS-EDA. To strengthen the exploration performance, an enhanced distribution model that considers the information of population and distribution is utilized to generate new candidates. Moreover, when the algorithm stagnates, the MLS mechanism will be activated to perform a local search and shrink the search scope. The performance of the MLS-EDA in addressing complex optimization problems is verified using the CEC 2014 and CEC 2017 testbeds with 30D, 50D and 100D tests. Several modern algorithms, including the top-performing methods in the CEC 2014 and CEC 2017 competitions, are considered as competitors. The competitive performance of our proposed MLS-EDA is discussed based on the comparison results.

INDEX TERMS Estimation of distribution algorithm, real-numerical optimization, CEC 2014, CEC 2017, evolutionary computation.

I. INTRODUCTION

The last few decades have seen tremendous progress in the field of evolutionary computation. As one of the most effective tools for solving NP-hard problems, evolutionary computation has received widespread attention and has been extensively researched worldwide. The main ideas of evolutionary computation derive from the laws of physics and chemistry, the activities and phenomena of natural organisms, and human behaviors. The algorithms are designed to solve optimization problems in accordance with such natural laws through various means. The traditional representative optimizers include the genetic algorithm (GA) [1], the neural network (NN) [2], particle swarm optimization (PSO) [3] and differential evolution (DE) [4]. Many experts and scholars have continuously devoted themselves to the study of evolutionary algorithms and have proposed numerous novel, well-established methods, such as the grey wolf optimizer (GWO) [5], ant lion optimizer (ALO) [6], teaching-learning-based optimization (TLBO) [7], virus colony search (VCS) [8], Harris hawks optimization (HHO) [9] and nuclear reaction optimization (NRO) [10]. Moreover, research on

evolutionary algorithms involves not only exploring new search frameworks but also improving and applying existing algorithms; this latter direction of algorithm development has also led to the emergence of a wide variety of methods, such as L-SHADE [11], CPI-JADE [12], DOLTLBO [13] and GEDGWO [14].

Estimation of distribution algorithm (EDA) [15], as one of the traditional evolutionary computation techniques, has undergone considerable development over the past two decades. In contrast to traditional evolutionary algorithms, which uses crossover, mutation and selection mechanisms, the EDA possesses a unique characteristic in that it estimates the probability distribution model of selected solutions and iteratively evolves the whole population. The Gaussian distribution model is typically used in EDA to solve problems in the continuous domain. According to the structure of the Gaussian probability model and the relationships between variables, EDAs can be classified into univariate [16], bivariate [17] and multivariate [15] models, and the competitive performance of multivariate Gaussian distribution model, which is the topic of our research, has been verified in various applications compared with the other two methods. However, when solving multimodal problems, an EDA cannot effectively capture the features of such a problem through an

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

estimated Gaussian distribution model. Moreover, the variances of the distribution in the slope directions may shrink rapidly in a later stage of optimization [18]. Thus, the traditional EDA tends to suffer from premature convergence and can easily become trapped in stagnation.

To overcome these deficiencies, many improvements to the EDA have been made, which can be categorized into three types: distribution model modifications, population diversity enrichment and algorithm hybridization. Based on the fact that the multivariate Gaussian distribution model is determined by its estimated mean and covariance matrix, studies were initially carried out on modifications to the distribution model. Grahl *et al.* [19] developed a correlation-triggered adaptive variance scaling (CT-AVS) strategy to scale the variances. Soon after, the same authors investigated another triggering method for adaptive variance scaling named standard deviation ratio (SDR) [20]. Based on these foundations, the well-known AMaLGaM [21] algorithm combining AVS, SDR and anticipated mean shift (AMS), was proposed [22]. Further studies were conducted by Ren and his teammates [23], who expanded the AMS strategy to consider the mean quality. Moreover, they explored a novel AVS strategy with anisotropic adjustment based on the local fitness landscape [18]. Their other work was EDA² [24], in which an archive is adopted to store more promising solutions to revise the distribution scope. Other representative EDA modifications of this type are the covariance matrix adaptation evolution strategy (CMA-ES) [25] and its variants. The CMA-ES has a complex framework that uses “rank-1” and “rank- μ ” update strategies. The IPOP-CMA-ES [26] is a promising CMA-ES variant that uses a restart strategy to enrich the population diversity. Currently, the NBIPOP-aCMA-ES [27], based on the BIPOP-CMA-ES [28] and active CMA-ES [29], is widely identified as the most efficient CMA-ES variant. Although the NBIPOP-aCMA-ES exhibits outstanding performance, it suffers from a relatively high computational burden due to its complex search mechanism [30].

Yuan reported that maintaining population diversity plays a key role in EDA performance [31]. Thus, the second method of avoiding premature convergence is to combine an EDA with diverse search mechanisms that can effectively enrich population diversity. Huang *et al.* [32] employed a simulated annealing method as a local search strategy in an EDA. In Miquélez’s work [33], the Bayesian classifier method and an EDA were combined to establish a probability model and solve a continuous optimization problem. The effectiveness of incorporating a regularized learning model into an EDA was studied by Karshenas *et al.* [34]. Copula theory [35] and a probabilistic graphical model [36] have also been utilized in an EDA and applied to samples. In RWGEDA, random walk strategies were incorporated into an EDA [37]. Moreover, the effects of the promising area detection technique and niching method on EDA performance improvement have been addressed by various studies [38]–[40].

Hybridizing EDA with other optimization algorithms to fully exploit their respective advantages is also regarded an

effective strategy for improving performance. Qi *et al.* [41] integrated an EDA with PSO to solve the water distribution problem. In three studies [42]–[44], DE has been utilized as a local search method to develop new hybrid algorithms. Sun and Gu [45] proposed a hybrid evolutionary algorithm combining an EDA and cuckoo search (CS) to solve scheduling problem.

Although all of the studies discussed above improved EDA performance from various perspective, several deficiencies still remain that require improvement:

- In most EDA variants, the available of population information is not fully exploited. First, the information of the current best solution in each iteration is not utilized. Furthermore, the role of promising solutions in enhancing the local search ability of an EDA has not received much attention. Moreover, the remaining solutions that are not utilized in constructing the distribution model are always neglected or abandoned. All these factors increase the computational cost.
- Developing a novel efficient mechanism for enriching the population diversity that does not rely on the distribution model, which can be achieved by taking advantage of the results derived in each generation, is still an urgent problem for EDA improvement.
- For a hybrid EDA, switching among sub-algorithms with different search mechanisms complicates the task of balancing the exploration and exploitation processes. Additionally, hybridization may lead to a more complex execution framework, which will increase the computational cost. Moreover, hybrid algorithms tend to have more parameters that require adjustment, which will reduce the parameter sensitivity of such an algorithm.

Both the deficiencies described above and the development prospects of EDAs in the field of evolutionary computation due to their model-based characteristics prompt us to propose new algorithms of this type. EDAs are currently playing an increasingly important role in algorithm development. From CEC 2016 to CEC 2018, most of the top algorithms were based on the use of distribution models for sampling, such as UMOEAII [46] in 2016, EBOwithCMAR [47] in 2017, and HSES [48] in 2018. In contrast to these promising EDA-based variants, we propose a novel nonhybrid EDA variant using a multi-leader search (MLS) mechanism in an eigen coordinate system, called the MLS-EDA. The novel characteristics and features of our MLS-EDA can be attributed to the following two mechanisms.

- For each individual in the population, an enhanced distribution model that considers population location differences is utilized to produce new candidates. This mechanism makes full use of the current population information to improve the distribution model and avoid wasted computation. Moreover, this mechanism can diversify the search scope, thus helping to maintain population diversity.
- When the algorithm stagnates, the capacity of the archive will be adaptively enlarged to store more current

leader solutions. For each individual, a leader solution in the archive is randomly selected to guide the individual performing the local search in a different dominant region. Thus, the MLS mechanism will be achieved. This mechanism can exploit multiple high-quality solutions to increase the diversity of the search scope and reduces the dependence on an ill-shaped distribution of solutions; thus, this mechanism exhibits strong exploration behavior when the algorithm stagnates.

The performance of our MLS-EDA in solving complex benchmarks is evaluated using the CEC 2014 and CEC 2017 testbeds with 30D, 50D and 100D tests, and the experimental results are compared with those of several other well-established algorithms, including the winner of the CEC 2014 test, L-SHADE, and the two top algorithms in CEC 2017, HSES and ELSHADE-SPACMA. Moreover, several modern EDAs, including VCS and RWGEDA, are employed to further demonstrate the efficiency of our modification.

The rest of this study is organized as follows: Section 2 presents a brief description of the original EDA and a detailed introduction to the MLS-EDA. In Section 3, the performance of the MLS-EDA is evaluated on the complex CEC 2014 and CEC 2017 test suites, and the results are statistically analyzed and compared. In the final section, the conclusions and prospects of this research are presented.

II. MLS-EDA

A. REVIEW OF THE BASIC EDA

The EDA is a model-based algorithm in which the solution vectors follow a Gaussian probability distribution model as follows:

$$G(\mu, C) = \frac{(2\pi)^{-\pi/2}}{(\det C)} \exp\left(-\frac{(X - \mu)^T (C)^{-1} (X - \mu)}{2}\right) \quad (1)$$

where

$$\mu = \frac{1}{|A|} \sum_{i=1}^{|A|} x_i, \quad x_i \in A \text{ and } A \subset X \quad (2)$$

$$C = \frac{1}{|A|} \sum_{i=1}^{|A|} (x_i - \mu)(x_i - \mu)^T, \quad x_i \in A \text{ and } A \subset X \quad (3)$$

The symbol A in (2) and (3) denotes a selected set of high-quality solutions. The EDA uses an estimated Gaussian distribution based on the selected superior solutions in set A to iteratively generate new solutions, thus driving the algorithm to complete the optimization process. In this way, the new generation is produced as follows:

$$x_i = \mu + y_i, \quad y_i \sim N(0, C) \quad (4)$$

In the standard EDA, the selected high-quality solutions are mainly distributed in the hyperellipsoid of the original probability distribution and tend to lie closer to the dominant region. The long axis of the probability distribution hyperellipsoid estimated from these solutions will be perpendicular

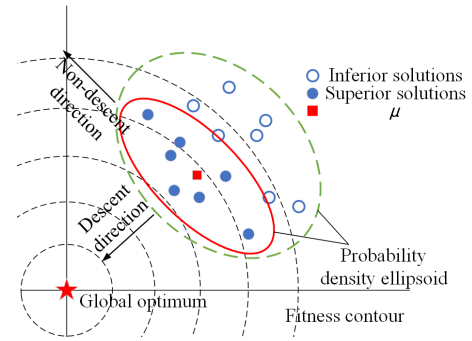


FIGURE 1. The variance in the non-descent direction is gradually enlarged, while the variance in the descent direction is gradually reduced.

to the descent direction of the function value. Fig. 1 shows an illustration of this phenomenon. Consequently, as the number of iterations increases, the main search direction of the algorithm become increasingly perpendicular to the descent direction of fitness, resulting in an ill-conditioned distribution of solutions, which allows the algorithm to easily fall into a local optimum. To overcome this defect, we have developed a modified EDA with multiple mechanisms, which is discussed in the following section.

B. DESCRIPTION OF THE MLS-EDA

The modifications proposed in the MLS-EDA consist of four components: a) a weighted maximum likelihood estimation (MLE) method to improve the mean point quality, b) a distribution enhancement strategy (DES) to diversify the distribution scope, c) an MLS mechanism to eliminate stagnation, and d) tan eigen coordinate framework to modify the direction of evolution. Further details are presented as follows.

1) WEIGHTED MAXIMUM LIKELIHOOD ESTIMATION METHOD

In the basic EDA, the MLE method is used to estimate the mean point, as shown in (2). In our MLS-EDA, a weighted MLE method is employed to calculate the weighted mean point, an approach that has been proven to be more efficient in many studies [14], [23], [25].

$$\mu = \sum_{i=1}^{|A|} \omega_i x_i, \quad x_i \in A \text{ and } A \subset X \quad (5)$$

where

$$\omega_i = \ln(|A| + 1) / \left(\sum_{i=1}^{|A|} (\ln(|A| + 1) - \ln(i)) \right) \quad (6)$$

In (6), the ω_i are the weight coefficients, arranged in descending order relative to the solution quality in set A . The weighted MLE method places greater emphasis on superior solutions in distribution estimation, thus improving the mean point quality. As done in most EDA variants, the top half of

the population is selected in set A [25], [37]; accordingly, $|A| = NP/2$, where NP denotes the population size.

2) DISTRIBUTION ENHANCEMENT STRATEGY

In (5), only the half of the solutions of the highest quality is employed to construct the distribution model, while the remaining inferior solutions are abandoned. To avoid this computational waste and to consider the differences in location between the superior and inferior solutions, a novel sampling method using a DES in an eigen coordinate system is explored.

First, each solution is judged as either a superior or inferior solution according to its fitness ranking in the population, which is calculated as

$$P_{\text{rank}}(i) = 1 + (1 - (f(x_i))_{\text{rank}}) / NP, \quad (f(x_i))_{\text{rank}} \in [1, 2, \dots, NP] \quad (7)$$

The i th solution will be regarded as a superior solution only if its fitness ranking $P_{\text{rank}}(i)$ is greater than 0.5; otherwise, it is an inferior solution. Since sampling is performed in an eigen coordinate framework, the solution vectors must be transformed from the normal coordinate system into the eigen coordinate system. The solution vectors in the eigen coordinates can be determined as follows:

$$\begin{cases} \mu^{(E)} = \mathbf{B}^T \mu \\ x_i^{(E)} = \mathbf{B}^T x_i \end{cases} \quad (8)$$

where the symbol E indicates a vector expressed in eigen coordinates. \mathbf{B} is the eigendirection matrix and obtained through the decomposition of \mathbf{C} , as shown in (9).

$$\mathbf{C} = (\mathbf{B}\mathbf{D})(\mathbf{B}\mathbf{D})^T \quad (9)$$

Generally, a superior solution is usually located closer to the global optimum point than the mean point, as illustrated in Fig. 1. Thus, the mean point can be enhanced by moving it towards the high-quality solution as follows:

$$\mu_j^{(E)} = \mu_j^{(E)} + r \cdot (x_{i,j}^{(E)} - \mu_j^{(E)}), \quad r \sim U(0, 1) \quad (10)$$

The subscript j in (10) denotes the j th dimension. Thus, a new candidate is produced as follows:

$$x_{i,j}^{(E)} = \mu_j^{(E)} + D_j y_{i,j}, \quad y_{i,j} \sim N(0, 1) \quad (11)$$

\mathbf{D} in (11) is a vector of the square root of the eigenvalues, i.e., $\mathbf{D} = (\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_{\text{dim}}})$, which is obtained as shown in (9). As shown in (10), the enhanced mean point contains information on both the distribution and the individual solution; thus, the distribution scope can be diversified, as shown in Fig. 2, which is helpful for solving multimodal problems. Moreover, because (10) is executed on each dimension individually, it can take advantage of the difference between the normal coordinates and the eigen coordinates, as presented in Fig. 2. Because the eigen coordinate system can eliminate the dependence among the different axes, the probable distribution of the shifted mean point in the

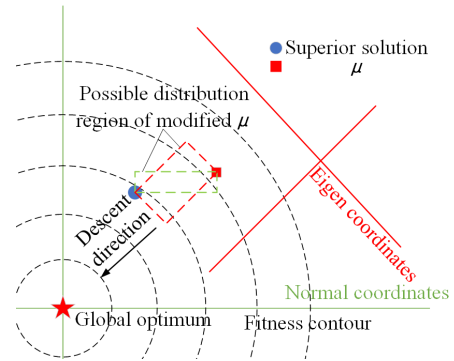


FIGURE 2. The enhanced mean point is obtained by moving the mean point towards a superior solution.

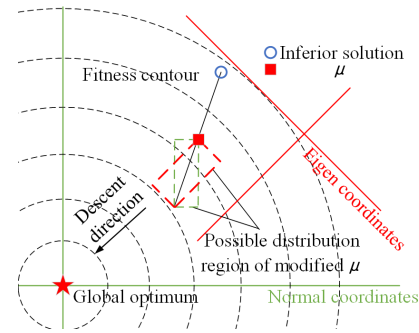


FIGURE 3. The enhanced mean point is obtained by moving the mean point away from the inferior solution.

eigen coordinates (in red) is more satisfactory than that in the normal coordinates (in green).

On the other hand, an inferior solution is usually located outside of the fitness contour across the mean point. Accordingly, a new candidate solution can be generated by enhancing the mean point by moving it away from such an inferior solution, as shown in (12) and (13).

$$\mu_j^{(E)} = \mu_j^{(E)} + r \cdot (\mu_j^{(E)} - x_{i,j}^{(E)}), \quad r \sim U(0, 1) \quad (12)$$

$$x_{i,j}^{(E)} = \mu_j^{(E)} + D_j y_{i,j}, \quad y_{i,j} \sim N(0, 1) \quad (13)$$

3) MULTI-LEADER SEARCH MECHANISM

Unlike the CMA-ES, the basic EDA has a poor ability to modify its ill-shaped distribution of solution. Thus, the basic EDA cannot easily eliminate local stagnation. To overcome this deficiency by enhancing the population diversity of the basic EDA, we propose the MLS mechanism, which is executed only when the algorithm stagnates. If the distribution model shows no change compared with the previous generation, i.e., the first half of the population is not updated, then the algorithm is considered to be trapped in stagnation and the MLS mechanism is activated.

The core idea of the MLS technique is to perform a local search based on multiple top solutions as leaders. In our MLS mechanism, an archive (denoted by \mathcal{S}) with an adaptive size is designed and employed to store the current dominant solutions. Initially, the size of \mathcal{S} is equal to 1, meaning that

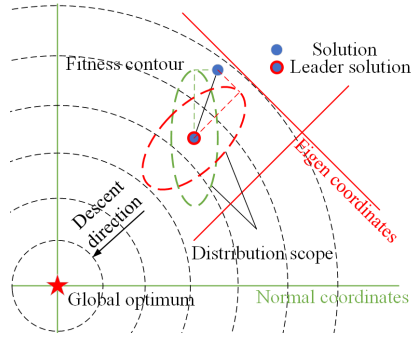


FIGURE 4. The MLS mechanism can generate an efficient local search around the selected leader.

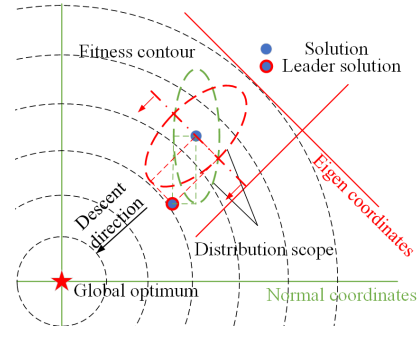


FIGURE 5. The MLS mechanism can generate an efficient mean point around at selected leader. $|z_i|$ is used to ensure that half of the distribution ellipsoid in the descent direction is retained.

the single current best solution is stored. When the algorithm stagnates for the first time, the number of solutions in archive S is increased by 1, meaning the two current best solutions are stored, and so on. When S reaches its predefined maximum size, its capacity will no longer be expanded. The next time that the algorithm stagnates, the size of the archive will be restored to the initial state and will not change, meaning that once again, only the single best solution is stored in S . The maximum size of S is set equal to three-times the dimensionality of the problem, as is discussed in the next section.

To sufficiently balance the exploration and exploitation performance, this mechanism incorporates two different search behaviors. If the i th solution is not selected in set S , i.e., if $P_{\text{rank}}(i) < 1 - |S|/NP$, then the MLS mechanism is executed around a selected dominant solution, as shown in (14).

$$\begin{aligned} \mathbf{x}_i^{(E)} &= \mathbf{x}_{\text{leader}}^{(E)} + \mathbf{y}_i \left(\mathbf{x}_{\text{leader}}^{(E)} - \mathbf{x}_i^{(E)} \right) + r_1 \cdot \mathbf{x}_{\text{leader}}^{(E)} - r_2 \cdot \mathbf{x}_i^{(E)}, \\ \mathbf{y}_i &\sim N(0, \mathbf{I}), \quad r_1, r_2 \sim N(0, 1), \quad \mathbf{x}_{\text{leader}} \in S \end{aligned} \quad (14)$$

In (14), $\mathbf{x}_{\text{leader}}$ is a leader solution and is randomly selected from archive S . The local search area in the eigen coordinate system is satisfactory, as illustrated in Fig. 4. As the number of instances of convergence stagnation increases, exploration is performed based on more dominant solutions in archive S . Because these dominant solutions may be located in different promising regions, the local search expressed in (14) can improve the exploration behavior of the algorithm in different dominant regions.

Otherwise, when the i th solution is selected in set S , i.e., when $P_{\text{rank}}(i) > 1 - |S|/NP$, the corresponding candidate is sampled using a shrunken distribution model. This alternative behavior of the MLS mechanism is executed as follows:

$$\begin{aligned} \mathbf{x}_i^{(E)} &= \mathbf{x}_i^{(E)} + |z_i| \cdot \left(\mathbf{x}_{\text{leader}}^{(E)} - \mathbf{x}_i^{(E)} \right) + \mathbf{D}' \mathbf{y}_i, \\ z_i &\sim N(0, 1), \quad \mathbf{y}_i \sim N(0, \mathbf{I}), \quad \mathbf{x}_{\text{leader}} \in S \end{aligned} \quad (15)$$

where

$$\mathbf{D}' = \mathbf{D} \cdot (1 - FE_s/FE_{s_{\text{max}}}) \quad (16)$$

The first two terms in (15) represent local exploitation around the i th solution using the location difference between

the i th solution itself and a randomly selected dominant solution. This search scope derived from the first two terms in (15) in the eigen coordinates is illustrated in Fig. 5. The absolute value of z_i is used as a truncation setting, which ensures that half of the search scope near the dominant area is preserved. In fact, this process also relies on the use of a leader solution for local search to generate the distribution mean point, thus enhancing the distribution and even the diversity of the population. As presented in (16), the elements of vector \mathbf{D} decrease as the number of evaluations increases, thereby reducing the search scope. Thus, in later search stages, the MLS mechanism focuses on a more precise local search around the randomly selected leader solution.

Both search behaviors expressed in (14) and (15) utilize a leader solution as the center around which to execute local exploitation. The critical difference is that the former behavior uses a location difference to generate new solutions, while the latter behavior uses a reduced distribution scope for sampling. In other words, both search behaviors in the MLS mechanism are designed to achieve the same end.

Finally, the newly obtained solution is transformed into the normal coordinates through rotation as follows:

$$\mathbf{x}_i = \mathbf{B} \cdot \mathbf{x}_i^{(E)} \quad (17)$$

To ensure the global convergence performance of the MLS-EDA, a greedy technique is utilized to preserve the superior solutions in both the current population and the newly produced population, thereby preventing the loss of high-quality solutions in the sire generation. The pseudocode for the MLS-EDA is presented below, and the execution process of the algorithm is visually summarized in Fig. 6.

C. COMPUTATIONAL TIME COMPLEXITY ANALYSIS OF THE MLS-EDA

The main factor that typically limits the application of EDA is the high computational cost, which is related to the dimensionality of the problem. In our MLS-EDA, the time complexity of updating the covariance matrix as expressed in (3), is $O(D^2 \cdot NP/2)$. Moreover, we use the Jacobi method to decompose the covariance matrix, as expressed in (9), with a time complexity of $O(D^3)$. In the population update process,

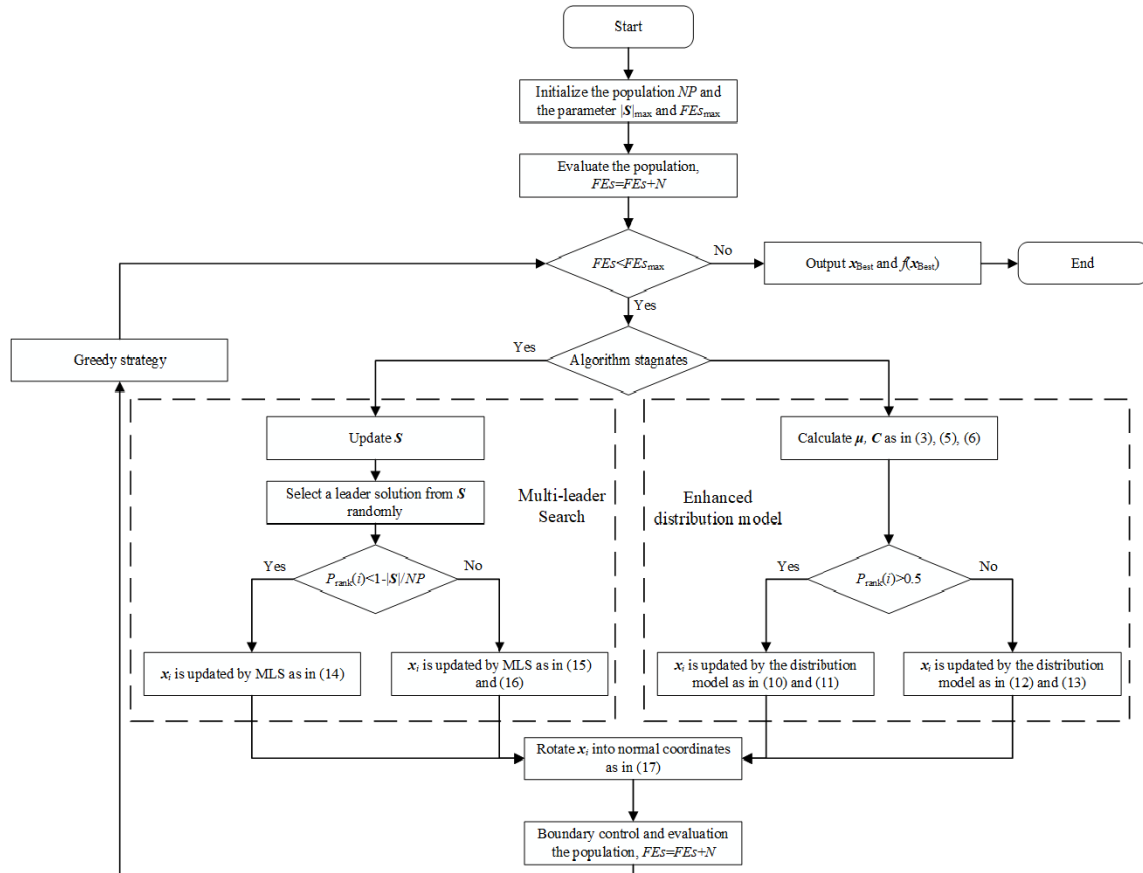


FIGURE 6. Flowchart of the MLS-EDA.

sampling is executed in each dimension for all individuals, thus leading to a computational cost of $O(NP \cdot D)$. Therefore, the maximum time complexity of the MLS-EDA is by $O\{D^2 \times \max(D, NP/2)\}$. Additionally, according to the execution process depicted in Fig. 6, the computational cost of updating the covariance matrix updating can be avoided when the algorithm stagnates. Under these conditions, our MLS-EDA has similar efficiency to the original EDA with a time complexity less than $O\{D^2 \times \max(D, NP/2)\}$ in each iteration.

III. EXPERIMENTAL STUDY USING MODERN CEC 2014 AND CEC 2017 TESTBEDS

There are no standard criteria for convincing verification of the real contributions of a novel algorithm in the field of evolutionary computation. We believe that three conditions are indispensable. First, comparison with advanced algorithms from various families is mandatory. Second, the use of a complete set of modern complex benchmarks, including the same conditions and parameter settings, is suggested. Finally, the advantages of the proposed algorithm should be properly specified. Thus, we employ the challenging CEC 2014 and CEC 2017 test suites to assess the performance of the MLS-EDA and carry out two comparisons. In the first

experiment, six promising algorithms from various families that participated in CEC 2014 are applied to demonstrate the superior performance of our proposed algorithm. In the second comparison, seven modern methods, including the top-performing algorithms in CEC 2017, are considered as competitors.

The CEC 2014 test suite contains 30 benchmarks (denoted by F1 to F30), while the CEC 2017 suite consists of 29 benchmarks (excluding F2). The tests in these two sets possess different characteristics and, accordingly, can be categorized into four groups. Further descriptions of these two test suites are provided in [49] and [50]. As recommended by the organizers, each benchmark should be independently run 51 times with the same maximum function evaluations (FES_{max}) to ensure a fair evaluation. The value of FES_{max} is set to $D \times 10,000$, where D denotes the dimensionality of the problem. In this paper, we carry out a full comparison using $30D$, $50D$ and $100D$ problems. The results are recorded in the form of error values between the derived optimal result and the global optimal result, where the global optimum is determined to be achieved with an error value less than $1e-08$. All simulations reported in this paper were run in the same hardware environment: a laptop (with a 2.20 GHz i7-8700HQ CPU and 16 GB of memory) and MATLAB 2018a was employed for coding and execution.

Algorithm MLS-EDA

```

1: Initialization:  $NP, S, FEs = 0$  and  $FEs_{max}$ ;
2: Produce the initial population  $X$  and evaluate,  $FEs = FEs + NP$ ;
3: Update the best solution  $x_{best}$  in the present generation;
4: While  $FEs \geq FEs_{max}$ , output  $f(x_{best})$  and terminate the search process; otherwise
5: Update the distribution model, including  $\mu$  and  $C$  using the half of the solutions of the highest quality in  $A$ , by means of (5), (6) and (3);
6: Calculate the eigenvalue decomposition of  $C$  using (9);
7: For each solution  $x_i$ 
  Calculate  $P_{rank}(i)$ ;
  Rotate  $x_i$  into the eigen coordinate system and determine  $x_i^{(E)}$  using (8);
  If the algorithm falls into stagnation
    Update the size of archive  $S$ ;
    If  $P_{rank}(i) < 1 - |S|/NP$ 
      Update  $x_i^{(E)}$  using the MLS behavior as expressed in (14);
    Else
      Update  $x_i^{(E)}$  using the MLS behavior as expressed in (15) and (16);
    End if
  Otherwise
    If  $P_{rank}(i) > 0.5$ 
      Update  $x_i^{(E)}$  using the enhanced mean as expressed in (10) and (11);
    Else
      Update  $x_i^{(E)}$  using the enhanced mean as expressed in (12) and (13);
    End if
  End if
  Rotate candidate  $x_i^{(E)}$  into the normal coordinate system by (17);
  Calculate the fitness  $f(x_i)$ ;
End for
9:  $FEs = FEs + NP$  and go to step 3;

```

In Appendix, we present the experimental results obtained on the CEC 2014 and CEC 2017 test suites, respectively, for 30D, 50D and 100D tests using the MLS-EDA with the population size set to $10 \cdot D$ and the maximum size of set S set to $3 \cdot D$. These statistical results, including the best, worst, median, mean and standard deviation (SD) error values, can be referenced in future studies. Moreover, the convergence performances achieved on these two test suites is illustrated in Appendices B and D. Since the values of parameters NP and $|S|_{max}$ in the MLE-EDA need to be determined, we analyze the optimal parameter settings for the MLS-EDA before presenting the comparisons with other techniques.

A. PARAMETER SENSITIVITY ANALYSIS OF THE MLS-EDA

The parameter sensitivity of an evolutionary algorithm is essential when addressing different problems. The MLS-EDA has two key parameters: the population size NP and the maximum size of set S . In this section, we present two experimental studies to investigate the optimal parameter values using the CEC 2014 30D test suite. All benchmarks are run 51 times with $FEs_{max} = 300,000$ to obtain the mean error value, and the results are presented for comparison in Tables 1 and 2.

In Table 1, the results for five population sizes, $NP = 6 \cdot D, 8 \cdot D, 10 \cdot D, 12 \cdot D$ and $14 \cdot D$, are given. The last row in Table 1 indicates the ranks of these five settings according to the Friedman test. The best performance is achieved by the algorithm with $NP = 10 \cdot D$. As the population size decreases or increases, the performance of the algorithm will decrease accordingly. The reason for this phenomenon is that a small population size cannot provide sufficient support to precisely capture the problem characteristics, whereas a larger population size will cause the number of iterations to decrease, thus compromising the convergence performance. Moreover, a larger population size can result in a greater computational cost since the maximum time complexity is $O\{D^2 \times \max(D, NP/2)\}$.

In Table 2, the rankings are assessed for settings of $|S|_{max} = 1, D, 2 \cdot D, 3 \cdot D, 4 \cdot D$ and $5 \cdot D$. When the value of $|S|_{max}$ is set to 1, this indicates that only the single best solution is considered for local exploitation in the MLS mechanism. According to the statistical results presented in the last row of Table 2, this setting leads to the worst behavior. In contrast, a larger size of S can significantly improve the algorithm's performance. All of the remaining five settings are acceptable, but $|S|_{max} = 3 \cdot D$ yields the most promising results, as indicated by its highest ranking.

TABLE 1. Mean error results achieved by the MSL-EDA with different population sizes on the CEC 2014 test suite for 30D problems.

No.	$NP=6\cdot D$	$NP=8\cdot D$	$NP=10\cdot D$	$NP=12\cdot D$	$NP=14\cdot D$
	Mean error value	Mean error value	Mean error value	Mean error value	Mean error value
01	4.40e+04	0.00e+00	0.00e+00	0.00e+00	0.00e+00
02	8.04e+06	0.00e+00	0.00e+00	0.00e+00	0.00e+00
03	1.81e+01	0.00e+00	0.00e+00	0.00e+00	0.00e+00
04	6.92e+01	1.94e-01	0.00e+00	0.00e+00	0.00e+00
05	2.00e+01	2.00e+01	2.00e+01	2.00e+01	2.00e+01
06	2.78e+00	2.33e-01	0.00e+00	3.20e-04	3.50e-01
07	4.88e-01	5.87e-04	0.00e+00	0.00e+00	0.00e+00
08	3.28e+01	2.75e+01	2.80e+01	3.01e+01	2.81e+01
09	3.23e+01	2.59e+01	2.34e+01	2.79e+01	2.76e+01
10	1.72e+03	1.89e+03	2.09e+03	2.29e+03	2.26e+03
11	2.64e+03	2.40e+03	2.38e+03	2.29e+03	2.43e+03
12	5.06e-02	2.65e-02	2.07e-02	3.50e-02	1.24e-01
13	1.72e-01	1.84e-01	2.09e-01	2.27e-01	2.27e-01
14	2.30e-01	2.09e-01	2.05e-01	2.02e-01	2.17e-01
15	3.63e+00	3.12e+00	3.05e+00	3.17e+00	3.18e+00
16	9.17e+00	9.30e+00	9.56e+00	9.53e+00	1.05e+01
17	5.75e+01	3.24e+01	2.39e+01	4.87e+01	7.67e+01
18	7.90e+00	6.10e+00	5.37e+00	5.69e+00	3.04e+00
19	2.37e+00	1.15e+00	1.38e+00	2.71e+00	3.45e+00
20	7.35e+00	4.39e+00	3.80e+00	4.29e+00	4.40e+00
21	3.93e+01	1.56e+01	1.04e+01	1.37e+01	1.59e+01
22	9.21e+01	1.05e+02	8.21e+01	1.10e+02	1.07e+02
23	2.00e+02	2.00e+02	2.00e+02	2.00e+02	2.00e+02
24	2.00e+02	2.01e+02	2.00e+02	2.01e+02	2.00e+02
25	2.00e+02	2.02e+02	2.00e+02	2.00e+02	2.00e+02
26	1.00e+02	1.00e+02	1.00e+02	1.00e+02	1.00e+02
27	3.80e+02	2.00e+02	2.00e+02	2.00e+02	2.06e+02
28	2.00e+02	2.00e+02	2.00e+02	2.28e+02	3.53e+02
29	2.55e+02	1.32e+02	1.18e+02	1.02e+02	1.00e+02
30	4.37e+02	3.81e+02	3.65e+02	3.99e+02	4.58e+02
Rankings	3.8500	2.7333	2.0333	2.9333	3.4500

The chi-square from the Friedman test is 29.1461, and the p-value is 7.2998e-06.

Overall, the optimal parameter settings of the MLS-EDA are $NP = 10\cdot D$ and $|S|_{\max} = 3\cdot D$. Notably, we have defined both of these parameters relative to the dimensionality of the problem, which is convenient for ensuring the use of parameter settings based on equivalent criteria when the algorithm is applied to solve problems of different dimensionalities. However, according to the no-free-lunch theorem, no single algorithm can be used to satisfactorily solve all problems. The optimal parameters demonstrated here are still not universally adequate, and additional application experience will be needed to further optimize these parameters.

B. MRE-EDA MODIFICATION COMPONENT ANALYSIS

In the previous subsection, the optimal parameter values are determined. In this part, we further analyze the influence on the algorithm's performance exerted by the different search behaviors driven by our modifications. As described in Section 2, our improvements mainly consist of two components, i.e., the DES and the MLS mechanism. Thus, three EDA variants are designed by removing one or both of these components, as presented in Table 3. These three variants, along with the MLS-EDA itself, are benchmarked for 30D problems using the CEC 2014 testbed. The same parameter settings are adopted, with 51 independent runs and the values of FES_{\max} fixed at 300,000. Table 4 presents the

statistical results, and the last row indicates the differences in performance among the four MLS-EDA variants. Clearly, our original MLS-EDA with both components achieves the best performance in this experiment. For the other algorithms, a lower ranking corresponds to a greater impact of the lost component on algorithm performance. Algorithm 2 and Algorithm 3 rank second and third, respectively. According to their rankings, there is little difference in performance between these two MLS-EDA variants. The poorest performance is exhibited by Algorithm 1, with both modifications removed. Specifically, as seen from the results in Table 4, the MLS mechanism improves the algorithm's performance in addressing the complex hybridization and composition benchmarks, while the DES significantly enhances the algorithm's ability to solve shifted and rotated problems. From these points of view, our proposed improvements substantially enhance the convergence performance of the basic EDA.

C. COMPARISON WITH SIX MODERN ALGORITHMS FROM DIFFERENT FAMILIES ON THE CEC 2014 TESTBED

In this subsection, six promising modern algorithms, namely, EMNA_g [15], RWGEDA [37], DOLTLBO [13], VCS [8], CPI-JADE [12] and L-SHADE [11], are considered as

TABLE 2. Mean error values achieved by the MSL-EDA with different maximum sizes of the set S on the CEC 2014 test suite for 30D problems.

No.	$ S _{\max}=1$	$ S _{\max}=D$	$ S _{\max}=2 \cdot D$	$ S _{\max}=3 \cdot D$	$ S _{\max}=4 \cdot D$	$ S _{\max}=5 \cdot D$
	Mean error value	Mean error value	Mean error value	Mean error value	Mean error value	Mean error value
01	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
02	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
03	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
04	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
05	2.00e+01	2.00e+01	2.00e+01	2.00e+01	2.00e+01	2.00e+01
06	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
07	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
08	2.95e+01	2.90e+01	2.86e+01	2.80e+01	2.29e+01	2.43e+01
09	2.89e+01	2.09e+01	2.83e+01	2.34e+01	2.69e+01	2.88e+01
10	2.09e+03	2.04e+03	2.15e+03	2.09e+03	2.05e+03	2.17e+03
11	2.45e+03	1.98e+03	2.40e+03	2.38e+03	2.38e+03	2.26e+03
12	1.99e-02	3.97e-02	2.17e-02	2.07e-02	2.10e-02	9.64e-03
13	2.15e-01	2.07e-01	2.03e-01	2.09e-01	2.06e-01	2.06e-01
14	1.97e-01	2.34e-01	2.23e-01	2.05e-01	2.26e-01	2.16e-01
15	2.85e+00	2.60e+00	2.79e+00	3.05e+00	2.44e+00	2.91e+00
16	9.59e+00	9.23e+00	9.60e+00	9.56e+00	9.59e+00	9.62e+00
17	2.77e+01	3.15e+01	2.25e+01	2.39e+01	4.01e+01	4.90e+01
18	6.88e+00	6.50e+00	5.54e+00	5.37e+00	5.35e+00	4.50e+00
19	7.07e-01	1.19e+00	1.23e+00	1.38e+00	1.71e+00	1.91e+00
20	4.92e+00	4.24e+00	4.28e+00	3.80e+00	4.74e+00	3.77e+00
21	1.13e+01	6.82e+00	6.23e+00	1.04e+01	1.05e+01	1.06e+01
22	1.13e+02	9.61e+01	1.04e+02	8.21e+01	8.55e+01	6.91e+01
23	2.00e+02	2.00e+02	2.00e+02	2.00e+02	2.00e+02	2.00e+02
24	2.02e+02	2.01e+02	2.01e+02	2.00e+02	2.01e+02	2.01e+02
25	2.02e+02	2.00e+02	2.00e+02	2.00e+02	2.00e+02	2.00e+02
26	1.00e+02	1.00e+02	1.00e+02	1.00e+02	1.00e+02	1.00e+02
27	2.53e+02	2.05e+02	2.00e+02	2.00e+02	2.00e+02	2.00e+02
28	3.52e+02	2.00e+02	2.00e+02	2.00e+02	2.00e+02	2.00e+02
29	1.19e+02	1.16e+02	1.15e+02	1.18e+02	1.19e+02	1.12e+02
30	3.74e+02	3.57e+02	3.56e+02	3.65e+02	3.64e+02	3.74e+02
Rankings	4.4167	3.2667	3.3500	3.1667	3.3833	3.4167

The chi-square of Friedman test is 14.1106, the p-value is 1.4922e-02.

competitors to demonstrate the efficiency of our MLS-EDA. EMNA_g is a basic multivariate Gaussian EDA, which we use to verify the efficiency of our modifications. RWGEDA is our recently developed EDA variant. DOLTLBO is a successful extension of TLBO. VCS is a well-established hybrid algorithm combining the CMA-ES and DE. CPI-JADE is a modification of JADE using a new framework. L-SHADE is an advanced DE algorithm whose variants have demonstrated championship performance in successive CEC competitions. We selected these six algorithms as competitors not only because they are relatively new members of their respective families but also because of their promising performance on the CEC 2014 benchmarks. The parameter settings play a key role in algorithm performance. To ensure fair comparisons, the parameter settings of the other six algorithms in this experiment are adopted in accordance with their respective source literature, as tabulated in Table 5. Each problem of different dimensionalities in this experiment is independently run 51 times to overcome the influence of randomness, with a fixed value of $FE_{S_{\max}} = D \times 10,000$, where $D = 30, 50$ and 100 . The simulation results for the tests with these three dimensionalities are presented in Tables 15 to 17, respectively, in Appendix E.

As presented in Table 15, the MLS-EDA, RWGEDA, CPI-JADE and L-SHADE perform robustly in addressing the

TABLE 3. MLS-EDA variants with different search components.

	DES	MLS
Algorithm 1	No	No
Algorithm 2	No	Yes
Algorithm 3	Yes	No
MLS-EDA	Yes	Yes

unimodal problems in the 30D test. L-SHADE performs better for solving the multimodal benchmarks, while our MLS-EDA shows significant advantages in solving the last two groups of functions.

According to the results for the 50D test in Table 16, the MLS-EDA and RWGEDA performs better in solving F1 to F3. For the multimodal functions F4 to F16, L-SHADE and MLS-EDA are the top two methods, with the best performance on seven and six benchmarks, respectively. Similar to the results in 30D test, the MLS-EDA outperforms the other algorithms in dealing with hybrid functions. However, VCS and DOLTLBO show competitive performance in solving the composite benchmarks.

Notably, the advantages of the MLS-EDA are reduced when solving high-dimensional problems. Nevertheless, from an overall perspective, the MLS-EDA maintains its competitiveness when dealing with multimodal problems.

TABLE 4. Rankings of the four MLS-EDA variants on the CEC 2014 test suite for 30D problems.

No.	Algorithm 1	Algorithm 2	Algorithm 3	MLS-EDA
	Mean error value	Mean error value	Mean error value	Mean error value
01	6.33e+06	1.20e+07	0.00e+00	0.00e+00
02	2.42e+09	1.69e+09	0.00e+00	0.00e+00
03	5.08e+03	2.99e+03	0.00e+00	0.00e+00
04	2.85e+02	2.99e+02	0.00e+00	0.00e+00
05	2.09e+01	2.02e+01	2.09e+01	2.00e+01
06	4.14e+00	4.36e+00	0.00e+00	0.00e+00
07	3.29e+01	2.15e+01	0.00e+00	0.00e+00
08	1.61e+02	4.32e+01	1.68e+02	2.80e+01
09	1.28e+02	5.19e+01	1.68e+02	2.34e+01
10	5.16e+03	1.85e+03	6.50e+03	2.09e+03
11	6.36e+03	2.90e+03	6.89e+03	2.38e+03
12	2.39e+00	1.05e-01	2.44e+00	2.07e-02
13	7.06e-01	4.41e-01	2.71e-01	2.09e-01
14	8.19e+00	6.78e+00	2.39e-01	2.05e-01
15	9.28e+00	4.27e+00	1.42e+01	3.05e+00
16	1.03e+01	8.30e+00	1.25e+01	9.56e+00
17	9.65e+02	4.32e+02	7.81e+02	2.39e+01
18	1.03e+02	9.46e+01	3.99e+01	5.37e+00
19	4.50e+00	4.75e+00	3.60e+00	1.38e+00
20	4.13e+01	3.33e+01	3.05e+01	3.80e+00
21	5.05e+02	5.39e+02	5.57e+02	1.04e+01
22	2.30e+02	1.42e+02	3.68e+02	8.21e+01
23	3.26e+02	2.00e+02	3.15e+02	2.00e+02
24	2.27e+02	2.10e+02	2.06e+02	2.00e+02
25	2.07e+02	2.00e+02	2.03e+02	2.00e+02
26	1.00e+02	1.00e+02	1.00e+02	1.00e+02
27	4.42e+02	2.00e+02	3.00e+02	2.00e+02
28	7.66e+02	2.00e+02	7.78e+02	2.00e+02
29	1.47e+03	1.62e+03	1.04e+02	1.18e+02
30	1.59e+03	1.68e+03	5.83e+02	3.65e+02
Rankings	3.3667	2.5833	2.7333	1.3167

The chi-square of the Friedman test is 42.8280, and the p-value is 2.6769e-09.

TABLE 5. Parameter settings of the seven algorithms in the CEC 2014 test.

Algorithms	Year	Parameter settings
MLS-EDA	-	$NP=10 \cdot D$, $\text{size}(\mathcal{S})_{\max}=3 \cdot D$
EMNA _g	2002	$NP=1000$, $\tau=0.35$ as in [15]
RWGEDA	2019	$NP=12 \cdot D$, $\tau=0.5$, $\alpha=0.05$, and $\beta=0.5$ as in [37].
DOLTLBO	2019	$NP=100$, $j_r=0.3$, $W=10$ as in [13].
VCS	2016	$NP=50$, $\lambda=0.5$, $\sigma=0.3$ as in [8].
CPI-JADE	2016	$NP=100$, $C_r=0.5$, $F=0.5$, $c=0.1$, $p=0.05$ as in [12].
L-SHADE	2014	$NP=r^{\text{Nimit}} \cdot D$, $NP_{\min}=4$, $r^{\text{Nimit}}=18$, $r^{\text{arc}}=2$, $p=0.1$, $H=5$ as in [11].

L-SHADE performs better in optimizing the first three types of benchmarks, whereas VCS and DOLTLBO are the two best-performing algorithms when faced with composite functions.

The Wilcoxon signed rank test and the Friedman test are carried out to statistically evaluate the efficacy of our proposed algorithm. The pairwise comparison results according to the Wilcoxon signed rank test with $\alpha = 0.05$ are presented in Table 6. In this table, “R+” denotes the magnitude by which the MLS-EDA can surpass a competitor, and “R-” represents the opposite effect. The rightmost column of Table 6 summarizes the overall comparisons, where the “+” entry indicates the number of cases in which the MES-EDA shows superior performance relative to the algorithm considered for comparison, the “-” entry indicates the number of cases of inferior performance, and the “~” entry indicates

TABLE 6. Comparison results of the Wilcoxon signed rank test ($\alpha = 0.05$).

	MLS-EDA	p-value	R+	R-	+/-/~
CEC 2014 30D test	vs. EMNA _g	6.89e-05	426	39	28/2/0
	vs. RWGEDA	3.42e-04	184	6	18/1/11
	vs. DOLTLBO	1.23e-05	325	0	25/0/5
	vs. VCS	2.03e-03	258	42	22/2/6
	vs. CPI-JADE	4.63e-03	287	64	22/4/4
	vs. L-SHADE	3.46e-01	183	117	15/9/6
CEC 2014 50D test	vs. EMNA _g	6.98e-06	451	14	29/1/0
	vs. RWGEDA	7.01e-04	213	18	17/4/9
	vs. DOLTLBO	1.72e-04	368	38	25/3/2
	vs. VCS	3.66e-02	276	102	21/6/3
	vs. CPI-JADE	4.79e-04	379	56	25/4/1
	vs. L-SHADE	1.35e-01	218	107	17/8/5
CEC 2014 100D test	vs. EMNA _g	1.08e-05	421	14	27/2/1
	vs. RWGEDA	8.00e-06	424	11	28/1/1
	vs. DOLTLBO	1.11e-02	356	109	22/8/0
	vs. VCS	9.10e-01	238	227	16/14/0
	vs. CPI-JADE	7.38e-01	202	233	15/14/1
	vs. L-SHADE	1.03e-01	142	293	11/18/1

the number of cases of similar performance. As shown by the statistical results of the 30D and 50D tests, our MLS-EDA outperforms all six competitors, with the “+” value always being the highest, but has a less significant advantage over

TABLE 7. Rankings of the seven algorithms on the CEC 2014 testbed based on the Friedman test ($\alpha = 0.05$).

Dimensionalities	MLS-EDA	EMNA _g	RWGEDA	DOLTLBO	VCS	CPI-JADE	L-SHADE
30D ^a	2.1333	6.4000	3.2333	5.3500	4.3667	3.7500	2.7667
Rank 50D ^b	2.2000	6.6333	2.9167	5.2167	3.7333	4.2167	3.0833
100D ^c	2.9667	6.4000	4.8833	4.6833	3.4667	3.2000	2.4000
SR	2.4333	6.4778	3.6778	5.0833	3.8556	3.7222	2.7500

- a. The chi-square is 9.15e+01, and the p-value is 1.46e-17.
- b. The chi-square is 9.17e+01, and the p-value is 1.32e-17.
- c. The chi-square is 7.65e+01, and the p-value is 1.89e-14.

TABLE 8. Mean time costs of the seven algorithms on the CEC 2014 testbed (in second).

Dimensionalities	MLS-EDA	EMNA _g	RWGEDA	DOLTLBO	VCS	CPI-JADE	L-SHADE
30D	6.1641	5.8709	5.8454	11.6035	7.6021	5.9852	5.5293
T 50D	17.1386	16.3110	17.9718	32.1196	20.9623	16.4544	16.4263
100D	87.9443	81.3754	88.8192	178.6610	112.4624	66.0291	84.9744
ST	37.0823	34.5191	37.5455	74.4947	47.0089	29.4896	35.6433

L-SHADE, with the p-value being greater than 0.05. In the 100D test, MLS-EDA surpasses EMNA_g, RWGEDA and DOLTLBO and performs similarly to VCS and CPI-JADE, but it is somewhat inferior to L-SHADE.

In addition to the pairwise comparisons above, a multiple comparison method, the Friedman test, is performed based on the mean values. Table 7 presents the rankings of the seven algorithms according to the Friedman test ($\alpha = 0.05$) for the three problem dimensionalities. The MLS-EDA ranks in first place on the 30D and 50D tests with the smallest ranking value, while L-SHADE ranks best on the 100D tests. The last row of this table shows the ranking performances of the algorithms on all three test sets in terms of a synthetically calculated score, denoted by SR. SR is calculated in accordance with the rank values in the first three rows of Table 7, as shown in (18).

$$SR = (Rank_{30D} + Rank_{50D} + Rank_{100D}) / 3 \quad (18)$$

Based on the SR value, the MLS-EDA is the best performing method among the seven algorithms. The CEC 2014 winner L-SHADE ranks second by a small margin. The remaining algorithms rank as follows, from best to worst: RWGEDA, CPI-JADE, VCS, DOLTLBO and EMNA_g. To further highlight the performance differences among the algorithms in this experiment, the post hoc Iman Davenport test was carried out based on the SR results. Further details about this method are provided in [14] and [51]. The critical difference (CD) value calculated in the post hoc test is 1.1998. A detailed illustration of the significant differences between different competitors in terms of the CD values according to the mean rank values is presented in Fig. 7. No significant difference exists between two algorithms if the average ranking difference between them is less than the CD value. As shown in Fig. 7, the MLS-EDA ranks first in this test and performs similarly to L-SHADE. Moreover, the MLS-EDA shows significantly superior performance compared with the remaining five algorithms, especially EMNA_g. This significant superiority

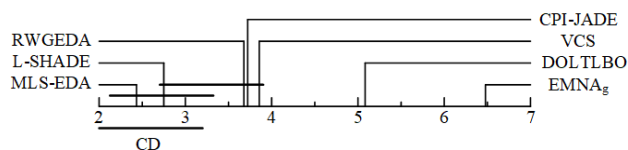


FIGURE 7. Multiple comparisons based on the post hoc Iman Davenport test.

demonstrates the value of the contributions of this work in modifying the basic EDA.

In addition to solution quality, execution efficiency is another important factor in algorithm evaluation. The average time cost of seven algorithms on the same platform for each benchmark are presented in Appendix F. Moreover, the average time consumption results for each algorithm on the tests of all three dimensionalities is calculated in Table 8. The symbol ST represents a time score calculated to compare the efficiency of these seven algorithms as expressed in (19).

$$ST = (T_{30D} + T_{50D} + T_{100D}) / 3 \quad (19)$$

In this experiment, CPI-JADE is the most efficient algorithm. The time consumption of MLS-EDA is higher than that of the basic EMNA_g because of its associated modifications, but the difference is small. The performance of RWGEDA is similar to that of the MLS-EDA, while VCS and DOLTLBO are the two algorithms with the lowest efficiency.

D. COMPARISON WITH SEVERAL WELL-ESTABLISHED METHODS ON THE CEC 2017 TESTBED

To adequately highlight the performance of our proposed method, tests on the CEC 2017 benchmarks with dimensionalities of 30D, 50D and 100D are also performed, and corresponding comparisons with advanced modern algorithms are presented in this subsection. The algorithm considered for comparison with the MLS-EDA here are EMNA_g, NRO [10], MRDE [52], RWGEDA [37], ELSHADE-SPACMA

TABLE 9. Parameter settings of the eight algorithms on the CEC 2017 test.

Algorithms	Year	Parameter settings
MLS-EDA	-	$NP=10 \cdot D$, $\text{size}(\mathbf{S})_{\max}=3 \cdot D$
EMNA _g	2002	$NP=1000$, $\tau=0.35$ as in [15]
NRO	2019	$NP=5 \cdot D$, $P_{fi}=0.75$, $P_{\beta}=0.1$, $\text{freq}=0.05$, $\alpha=0.01$, $\beta=1.5$ as in [10]
MRDE	2019	$NP=gs \cdot gn$, $gs=3$, $gn=25$, $\text{Max}_{\text{imp}}=2$, $\text{Max}_{\text{stag}}=3$, $\text{Min}_{ps}=NP/3$, $\text{Max}_{ps}=NP$ as in [52]
RWGEDA	2019	$NP=12 \cdot D$, $\tau=0.5$, $\alpha=0.05$, and $\beta=0.5$ as in [37]
ELSHADE-SPACMA	2018	$NP=r^{\text{Ninit}} \cdot D$, $NP_{\min}=4$, $r^{\text{Ninit}}=18$, $r^{\text{arc}}=1.4$, $p^{\text{init}}=0.3$, $p^{\text{min}}=0.15$ as in [53]
ACoS-JADE	2018	$NP=8 \cdot D$, $\text{Size}(\mathbf{A})=3 \cdot NP$, $\mu=AS/2$, $c=0.1$, $p=0.05$, $CR_m=0.5$, $F_m=0.5$, $\varepsilon=0.05$, $\eta=0.1$ as in [54]
HSES	2018	$NP=\lfloor 3 \cdot \ln D \rfloor + 80$, $M=450$, $N=360$, $cc=0.96$, $I=20$ as in [48]

TABLE 10. Comparison results of the WILCOXON signed rank test ($\alpha = 0.05$).

	MLS-EDA	p-value	R+	R-	+/-/≠
CEC 2017 30D test	vs. EMNA _g	2.56e-06	435	0	29/0/0
	vs. NRO	1.67e-03	40	260	3/22/4
	vs. MRDE	1.32e-04	326	25	22/4/3
	vs. RWGEDA	6.06e-05	250	3	21/2/6
	vs. ELSHADE-SPACMA	5.03e-01	160	116	13/10/6
	vs. ACoS-JADE	3.88e-05	294	6	23/1/5
	vs. HSES	8.31e-01	145	131	10/14/5
CEC 2017 50D test	vs. EMNA _g	6.53e-06	426	9	26/3/0
	vs. NRO	8.29e-01	193.5	212.5	16/12/1
	vs. MRDE	3.46e-05	409	26	26/3/0
	vs. RWGEDA	1.67e-05	345	6	24/2/3
	vs. ELSHADE-SPACMA	6.07e-01	168	132	15/9/5
	vs. ACoS-JADE	9.33e-06	350	1	25/1/3
	vs. HSES	8.65e-02	108	243	7/19/3
CEC 2017 100D test	vs. EMNA _g	2.56e-06	435	0	29/0/0
	vs. NRO	6.60e-04	375	60	24/5/0
	vs. MRDE	4.80e-06	429	6	27/2/0
	vs. RWGEDA	5.90e-06	427	8	28/1/0
	vs. ELSHADE-SPACMA	3.31e-03	317	61	18/10/1
	vs. ACoS-JADE	8.85e-06	372	6	26/3/0
	vs. HSES	2.75e-01	149	229	9/20/0

[53], ACoS-JADE [54] and HSES [48]. NRO is a promising physical-based optimizer that performed well in the CEC 2017 test, MRDE is a newly developed DE variant, RWGEDA and ACoS-JADE are two covariance-matrix-based algorithms, and ELSHADE-SPACMA and HSES are the two top performers in the CEC 2018 competitions. The parameter settings of each algorithm are adopted in accordance with the respective original work, as shown in Table 9. All 29 benchmarks in the CEC 2017 testbed are evaluated 51 times with a predefined evaluation number of $D \times 10,000$, where $D = 30, 50$ and 100 . The statistical data, including the mean and SD error values, are presented in Tables 19 to 21, in Appendix G.

As observed in Table 19, the MLS-EDA outperforms the basic EMNA_g in solving the unimodal benchmarks. However, NRO shows great superiority on the 30D test, achieving the best mean values on almost all of the benchmarks except F12, F25 and F28. By contrast, for the 50D test, as shown

in Table 20, NRO is significantly affected by the problem dimensionality, whereas HSES achieves first rank in more cases. Our MLS-EDA performs better on F1, F3, F9, F27 and F28. When addressing the 100D problems, HSES and our MLS-EDA are the two best-performing algorithms, as presented in Table 21.

As in the previous experiment, the Wilcoxon signed rank test is carried out to evaluate the pairwise differences between the MLS-EDA and the other competitors, as shown in Table 10. The symbols in this table have same meanings as in Table 6. In the 30D test, our MLS-EDA exhibits performance similar to that of ELSHADE-SPACMA and HSES and surpasses EMNA_g, MRDE, RWGEDA and ACoS-JADE, but it is outperformed by NRO. In the 50D and 100D tests, the MLS-EDA achieves a higher ‘+’ count than any of the other algorithms except HSES.

Furthermore, the Friedman test is carried out to rank these eight algorithms according to their performance in the tests of all three dimensionalities. Table 11 presents the ranking results based on the Friedman test ($\alpha = 0.05$). According to these ranking values, NRO is the winner on the 30D test and HSES performs the best on the 50D and 100D tests. Although our MLS-EDA is not the best-performing algorithm on any of the three test sets, it shows robust performance earning second place in all three competitions. Moreover, when the SR calculation is used to determine the overall ranking of the eight algorithms, the winner of the CEC 2018 competition, HSES similarly ranks the best in this experiment. Our MLS-EDA lags behind HSES by only a small margin, ranking second. Benefiting from its out performance on 30D test, NRO ranks third. The other algorithms rank as follows, from best to worst: ELSHADE-SPACMA, RWGEDA, ACoS-JADE, MRDE and EMNA_g.

Fig. 8 illustrates the significant differences between the different algorithms based on the Iman Davenport test. The CD value is 1.3219. Clearly, the MLS-EDA shows little difference from HSES, NRO and ELSHADE-SPACMA but surpasses RWGEDA, MRDE, ACoS-JADE and EMNA_g.

Although HSES achieves the best overall score on the benchmarks in the CEC 2017 test suite, it is not the most efficient method according to the execution time costs reported in Appendix H. Table 12 shows the average amounts of time required by these eight algorithms to solve the benchmarks of different dimensionalities. The differences between the efficiencies of the various algorithms can be more clearly

TABLE 11. Rankings of the eight algorithms on the CEC 2017 testbed based on the Friedman test ($\alpha = 0.05$).

	Dimensions	MLS-EDA	EMNA _g	NRO	MRDE	RWGEDA	ELSHADE-SPACMA	ACoS-JADE	HSES
Rank	30D ^a	3.1724	7.9655	1.8966	5.2069	4.5517	3.8448	5.7759	3.5862
	50D ^b	2.9310	7.5172	3.4655	5.7586	4.8103	2.9655	5.9310	2.6207
	100D ^c	2.4310	7.7586	3.9655	5.4138	5.7931	2.9828	5.6724	1.9828
SR	2.8448	7.7471	3.1092	5.4598	5.0517	3.2644	5.7931	2.7299	

- a. The chi-square is 1.28e+02, and the p-value is 1.81e-24.
- b. The chi-square is 1.10e+02, and the p-value is 7.59e-21.
- c. The chi-square is 1.34e+02, and the p-value is 8.17e-26.

TABLE 12. Mean time costs of the eight algorithms on the CEC 2017 testbed (in second).

	Dimensions	MLS-EDA	EMNA _g	NRO	MRDE	RWGEDA	ELSHADE-SPACMA	ACoS-JADE	HSES
T	30D	3.7701	3.4623	6.1233	9.9733	3.8259	3.7777	6.5542	4.1022
	50D	9.8944	9.4636	16.5100	21.1926	9.9933	10.2285	15.1153	13.1574
	100D	63.8625	59.3155	58.3809	86.6279	67.5865	66.3098	78.7048	114.9892
ST	25.8423	24.0805	27.0047	39.2646	27.1352	26.7720	33.4581	44.0829	

TABLE 13. Error values obtained with the MLS-EDA on the CEC 2014 30D, 50D and 100D tests.

No.	Types	CEC 2014 30D					CEC 2014 50D					CEC 2014 100D				
		Best	Worst	Median	Mean	SD	Best	Worst	Median	Mean	SD	Best	Worst	Median	Mean	SD
01	Unimodal Functions	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	2.04e-02	9.72e-02	2.69e-02	4.42e-02	3.12e-02
02		0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	5.01e-01	8.43e-01	6.06e-01	6.48e-01	1.29e-01
03		0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	5.83e-07	1.03e-06	8.43e-07	8.38e-07	1.78e-07
04	Multimodal Functions	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.84e-03	1.08e+00	5.53e-01	5.59e-01	4.00e-01	1.43e+02	1.43e+02	1.43e+02	1.43e+02	3.97e-02
05		2.00e+01	2.02e+01	2.00e+01	2.00e+01	3.69e-02	2.00e+01	2.01e+01	2.00e+01	2.00e+01	4.42e-02	2.00e+01	2.00e+01	2.00e+01	2.00e+01	1.86e-04
06		0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.61e-04	1.61e-05	3.76e-05	6.27e-05	3.61e+00	8.96e+00	6.65e+00	6.69e+00	1.57e+00
07		0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	2.74e-07	4.59e-07	3.99e-07	4.04e-07	6.76e-08
08		1.59e+01	4.88e+01	2.89e+01	2.80e+01	8.28e+00	2.49e+01	6.07e+01	3.28e+01	3.71e+01	1.34e+01	7.52e+02	8.39e+02	7.76e+02	7.84e+02	2.76e+01
09		1.09e+01	4.68e+01	2.29e+01	2.34e+01	8.23e+00	2.39e+01	6.47e+01	3.33e+01	3.63e+01	1.48e+01	7.04e+02	7.86e+02	7.39e+02	7.44e+02	2.57e+01
10		2.80e+02	3.98e+03	2.02e+03	2.09e+03	8.08e+02	4.04e+03	6.39e+03	5.00e+03	5.05e+03	9.03e+02	9.88e+03	1.50e+04	1.28e+04	1.25e+04	1.32e+03
11		8.12e+02	4.09e+03	2.46e+03	2.38e+03	7.34e+02	2.62e+03	6.11e+03	4.41e+03	4.38e+03	1.11e+03	1.14e+04	1.34e+04	1.25e+04	1.24e+04	6.97e+02
12		6.91e-08	3.36e-01	6.18e-03	2.07e-02	5.74e-02	2.95e-03	6.82e-02	8.22e-03	2.36e-02	2.82e-02	3.94e-03	1.29e-01	7.35e-03	1.91e-02	3.68e-02
13		1.43e-01	3.00e-01	2.04e-01	2.09e-01	3.79e-02	3.14e-01	3.81e-01	3.33e-01	3.42e-01	2.60e-02	4.01e-01	5.33e-01	4.61e-01	4.70e-01	5.94e-02
14		9.32e-02	3.01e-01	2.02e-01	2.05e-01	4.13e-02	2.26e-01	2.67e-01	2.55e-01	2.51e-01	1.65e-02	2.15e-01	3.47e-01	2.74e-01	2.78e-01	4.18e-02
15		1.72e+00	5.98e+00	2.98e+00	3.05e+00	8.12e-01	3.35e+00	6.47e+00	4.44e+00	4.77e+00	1.19e+00	5.58e+01	6.96e+01	6.50e+01	6.38e+01	4.12e+00
16		7.21e+00	1.14e+01	9.63e+00	9.56e+00	8.45e-01	1.86e-01	2.09e-01	1.94e-01	1.94e-01	8.06e-01	4.19e-01	4.50e-01	4.35e-01	4.35e-01	8.64e-01
17		6.86e+00	1.44e+02	2.01e+01	2.39e+01	1.93e+01	1.25e+02	5.85e+02	2.04e+02	2.50e+02	1.70e+02	6.18e+03	7.55e+03	6.52e+03	6.70e+03	4.61e+02
18		4.38e-01	1.12e+01	5.27e+00	5.37e+00	2.37e+00	1.53e+00	8.84e+00	6.52e+00	6.08e+00	2.42e+00	2.36e+02	3.25e+02	2.64e+02	2.70e+02	2.34e+01
19		2.95e-01	3.96e+00	1.21e+00	1.38e+00	8.71e-01	6.92e+00	9.29e+00	8.40e+00	8.17e+00	9.59e-01	9.18e+01	9.34e+01	9.21e+01	9.23e+01	5.20e-01
20	Hybrid Function	1.00e+00	7.30e+00	3.61e+00	3.80e+00	1.67e+00	4.90e+00	1.04e+01	7.58e+00	7.92e+00	2.16e+00	1.46e+02	2.18e+02	1.81e+02	1.82e+02	1.95e+01
21	5.77e-01	1.36e+02	6.37e+00	1.04e+01	2.04e+01	1.35e+02	5.27e+02	2.46e+02	2.76e+02	1.43e+02	3.35e+03	4.34e+03	3.89e+03	3.85e+03	2.89e+02	
22	2.28e+01	1.91e+02	3.76e+01	8.21e+01	6.22e+01	6.93e+01	2.89e+02	2.13e+02	2.03e+02	8.58e+01	9.73e+02	2.59e+03	1.61e+03	1.75e+03	5.17e+02	
23	Composition Functions	2.00e+02	2.00e+02	2.00e+02	2.00e+02	0.00e+00	2.00e+02	2.00e+02	2.00e+02	2.00e+02	1.50e-10	3.48e+02	3.48e+02	3.48e+02	3.48e+02	1.24e-03
24		2.00e+02	2.20e+02	2.00e+02	2.01e+02	3.81e+00	2.00e+02	2.00e+02	2.00e+02	2.00e+02	1.84e-07	3.77e+02	3.87e+02	3.83e+02	3.82e+02	2.94e+00
25		2.00e+02	2.00e+02	2.00e+02	2.00e+02	2.60e-11	2.05e+02	2.05e+02	2.05e+02	2.05e+02	1.09e-01	2.15e+02	2.17e+02	2.16e+02	2.15e+02	7.25e-01
26		1.00e+02	1.00e+02	1.00e+02	1.00e+02	3.89e-02	1.00e+02	1.00e+02	1.00e+02	1.00e+02	5.58e-02	1.01e+02	2.00e+02	2.00e+02	1.76e+02	4.34e+01
27		2.00e+02	2.00e+02	2.00e+02	2.00e+02	0.00e+00	3.00e+02	3.00e+02	3.00e+02	3.00e+02	4.19e-02	3.97e+02	5.18e+02	4.58e+02	4.57e+02	4.11e+01
28		2.00e+02	2.00e+02	2.00e+02	2.00e+02	6.79e-08	2.08e+02	1.11e+03	1.00e+03	8.93e+02	3.39e+02	1.53e+03	4.24e+03	3.29e+03	3.12e+03	1.00e+03
29		1.00e+02	1.61e+02	1.00e+02	1.18e+02	2.34e+01	3.24e+02	3.48e+02	3.42e+02	3.39e+02	9.00e+00	7.02e+02	7.40e+02	7.10e+02	7.12e+02	9.73e+00
30		3.25e+02	4.43e+02	3.58e+02	3.65e+02	2.84e+01	8.05e+03	8.74e+03	8.72e+03	8.61e+03	2.73e+02	5.97e+03	7.08e+03	6.47e+03	6.48e+03	3.87e+02

seen from the ST results. The basic EMNA_g requires the least time, while the MLS-EDA performs more efficiently than any of the other methods except EMNA_g. HSES ranks last with a substantially greater time cost. These findings also reflect a bottleneck problem hindering the development of new algorithms. Better algorithm performance is typically achieved by means of a more complex algorithm structure, which will cause more time to be required to run the algorithm. However, our MLS-EDA shows advantages in both accuracy and speed. Thus, our MLS-EDA, with its new search mechanisms, exhibits competitive performance on the CEC 2017 test suite.

Through the above two experiments, the superiority of the MLS-EDA has been fully demonstrated. Here, we clarify the difference between the MLS-EDA and RWGEDA. These two algorithms have similar execution frameworks, but their search mechanisms are different. In RWGEDA, the random

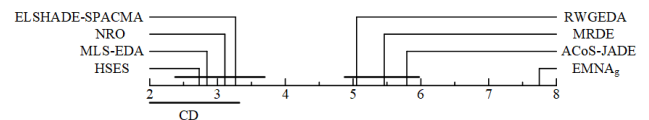


FIGURE 8. Multiple comparisons based on the post hoc Iman Davenport test.

walk strategies only utilize the best solution to conduct the local search. This feature is not appropriate for our MLS-EDA, as confirmed by its weakness as demonstrated in the above study on parameter sensitivity. Instead, the MLS mechanism that employs more high-quality solutions is more efficient. Furthermore, the newly developed MLS-EDA shows more promising performance than RWGEDA does on the CEC 2014 and CEC 2017 tests in terms of both accuracy and efficiency. In some sense, the MLS-EDA is a superior development of the RWGEDA.

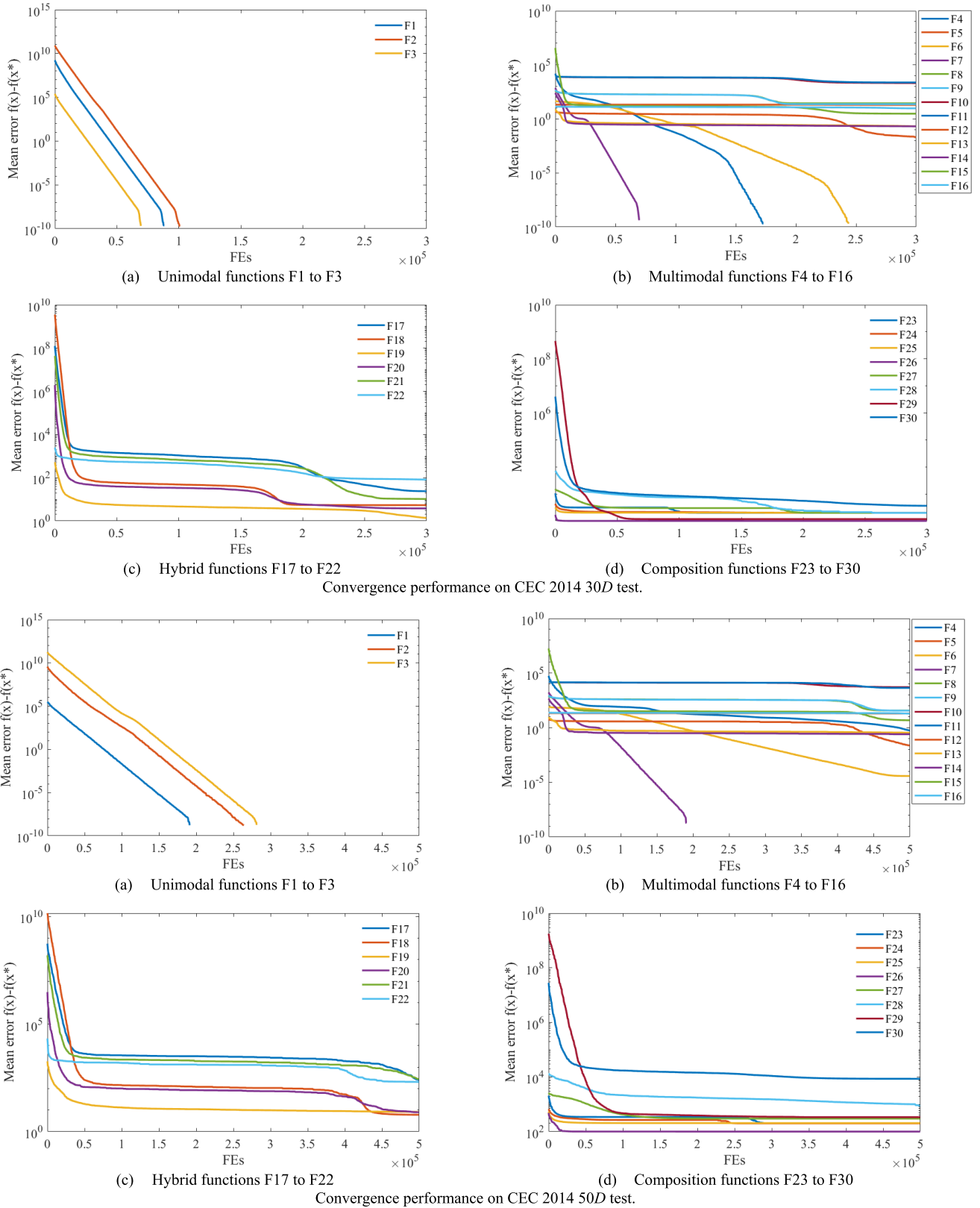


FIGURE 9. Convergence performance of the MLS-EDA on the CEC 2014 30D, 50D and 100D tests.

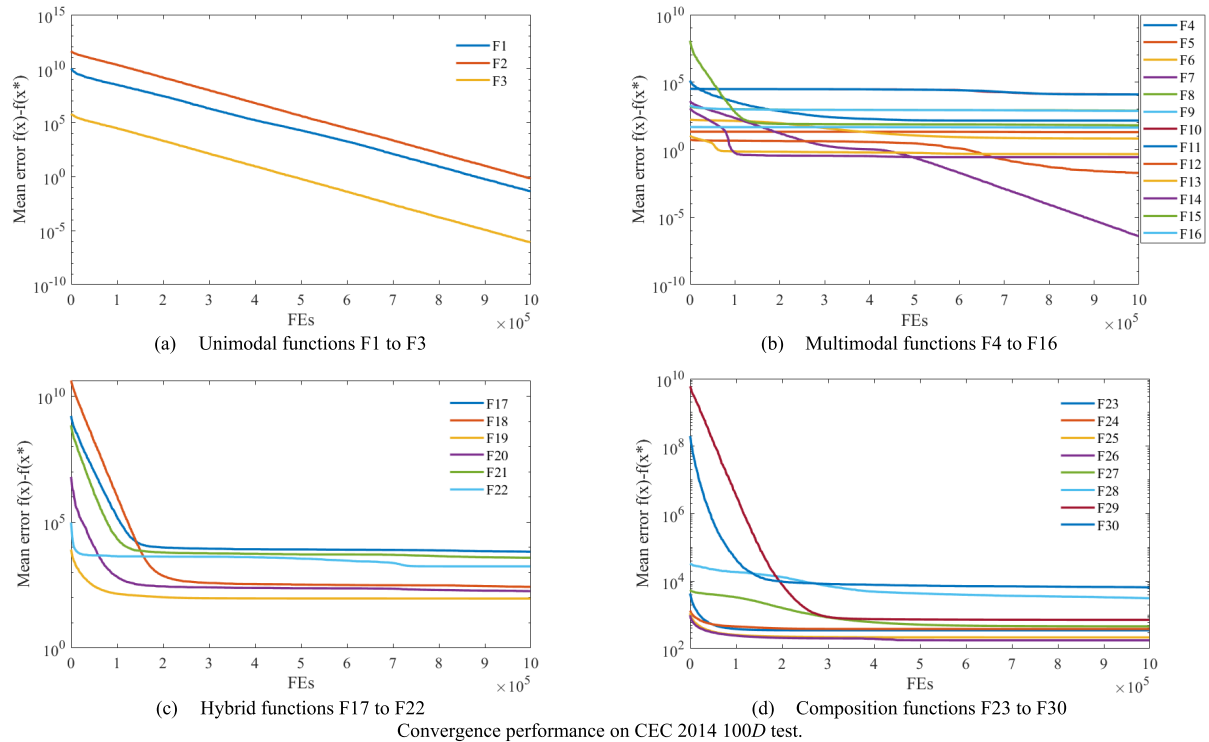


FIGURE 9. (Continued) Convergence performance of the MLS-EDA on the CEC 2014 30D, 50D and 100D tests.

TABLE 14. Error values obtained with the MLS-EDA ON the CEC 2017 30D, 50D and 100D tests.

No.	Types	CEC 2017 30D					CEC 2017 50D					CEC 2017 100D				
		Best	Worst	Median	Mean	SD	Best	Worst	Median	Mean	SD	Best	Worst	Median	Mean	SD
01	Unimodal Functions	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.10e-02	1.98e-02	1.48e-02	1.31e-02	1.69e-03
03		0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	9.82e-07	2.20e-06	1.23e-06	1.24e-06	5.93e-07
04		5.06e+01	5.61e+01	5.06e+01	5.14e+01	2.10e+00	0.00e+00	1.26e+02	2.85e+01	3.86e+01	3.57e+01	1.92e+02	2.12e+02	2.12e+02	2.06e+02	9.09e+00
05	Multimodal Functions	8.95e+00	3.88e+01	2.79e+01	2.60e+01	9.97e+00	1.49e+01	8.95e+01	3.58e+01	3.73e+01	1.30e+01	1.99e+01	2.98e+01	2.39e+01	2.41e+01	2.93e+00
06		0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	2.38e-06	4.02e+03	1.03e-04	6.62e-04	1.13e-03	4.03e-03	4.49e-03	4.19e-03	4.19e-03	1.52e-04
07		4.60e+01	6.65e+01	5.66e+01	5.64e+01	7.21e+00	6.31e+01	1.34e+02	8.03e+01	8.27e+01	1.38e+01	1.20e+02	1.54e+02	1.31e+02	1.32e+02	1.15e+01
08		1.19e+01	4.88e+01	2.19e+01	2.53e+01	1.28e+01	1.89e+01	6.67e+01	3.38e+01	3.74e+01	1.13e+01	1.79e+01	2.59e+01	2.19e+01	2.19e+01	2.72e+00
09		0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
10		1.48e+03	3.23e+03	2.75e+03	2.64e+03	5.41e+02	2.37e+03	7.47e+03	4.67e+03	4.69e+03	1.28e+03	8.12e+02	2.57e+03	1.61e+03	1.66e+03	5.92e+02
11		1.99e+00	1.39e+01	5.97e+00	6.68e+00	4.05e+00	2.13e+01	3.31e+01	2.51e+01	2.54e+01	2.63e+00	2.07e+01	1.65e+02	3.06e+01	5.68e+01	4.96e+01
12	Hybrid Function	2.78e-01	4.00e+00	3.93e+00	2.43e+00	1.95e+00	6.25e+01	1.49e+03	5.19e+02	6.43e+02	4.08e+02	2.85e+03	4.49e+03	3.18e+03	3.32e+03	5.23e+02
13		1.99e+00	2.00e+01	1.59e+01	1.17e+01	7.65e+00	5.12e+00	8.70e+01	1.42e+01	2.12e+01	1.89e+01	2.00e+01	1.28e+02	8.52e+01	7.48e+01	3.95e+01
14		2.98e+00	3.21e+01	9.95e+00	1.27e+01	1.13e+01	9.69e+00	4.17e+01	2.98e+01	3.01e+01	5.79e+00	2.19e+01	2.87e+01	2.48e+01	2.46e+01	2.29e+00
15		1.44e+00	9.15e+00	3.30e+00	3.78e+00	2.60e+00	1.82e+01	2.50e+01	2.13e+01	2.15e+01	1.76e+00	1.93e+01	8.96e+01	7.81e+01	5.98e+01	2.92e+01
16		7.60e+00	4.50e+02	2.40e+01	1.07e+02	1.60e+02	1.27e+02	1.26e+03	4.27e+02	4.42e+02	2.66e+02	7.19e+02	9.75e+02	9.03e+02	8.92e+02	1.11e+02
17		2.85e+01	5.46e+01	3.78e+01	3.89e+01	8.30e+00	1.47e+02	9.92e+02	4.14e+02	4.36e+02	1.80e+02	5.42e+02	6.64e+02	5.66e+02	5.94e+02	5.33e+01
18		4.70e-01	2.27e+01	2.07e+01	1.56e+01	1.04e+01	2.14e+01	2.38e+01	2.16e+01	2.20e+01	7.00e-01	2.13e+01	2.72e+01	2.52e+01	2.50e+01	1.80e+00
19		2.96e+00	4.30e+00	3.20e+00	3.47e+00	5.16e-01	8.30e+00	1.94e+01	1.21e+01	1.21e+01	2.11e+00	2.13e+01	2.95e+01	2.46e+01	2.48e+01	2.48e+00
20		1.13e+00	1.52e+02	2.28e+01	3.58e+01	5.34e+01	6.07e+01	9.43e+02	1.99e+02	2.56e+02	1.76e+02	2.83e+01	2.90e+02	3.23e+01	7.42e+01	9.14e+01
21		Composition Functions	2.22e+02	2.33e+02	2.22e+02	2.22e+02	6.09e+00	2.21e+02	2.69e+02	2.35e+02	2.36e+02	1.01e+01	2.37e+02	2.52e+02	2.47e+02	2.46e+02
22	1.00e+02		1.00e+02	1.00e+02	1.00e+02	0.00e+00	1.00e+02	6.38e+03	1.00e+02	7.34e+02	1.77e+03	1.81e+03	3.01e+03	2.48e+03	2.44e+03	3.58e+02
23	3.55e+02		3.77e+02	3.59e+02	3.63e+02	8.92e+00	4.08e+02	4.63e+02	4.36e+02	4.35e+02	1.29e+01	5.46e+02	5.82e+02	5.59e+02	5.60e+02	1.03e+01
24	4.18e+02		4.60e+02	4.25e+02	4.30e+02	1.44e+01	4.66e+02	5.27e+02	5.02e+02	5.02e+02	1.13e+01	8.01e+02	8.91e+02	8.65e+02	8.56e+02	3.07e+01
25	3.87e+02		3.87e+02	3.87e+02	3.87e+02	1.54e-02	4.55e+02	5.63e+02	4.80e+02	4.82e+02	2.35e+01	5.77e+02	7.62e+02	7.03e+02	6.84e+02	5.47e+01
26	6.54e+02		1.06e+03	9.49e+02	9.15e+02	1.34e+02	7.95e+02	1.35e+03	1.13e+03	1.12e+03	1.24e+02	2.45e+03	2.84e+03	2.74e+03	2.70e+03	1.29e+02
27	4.67e+02		4.74e+02	4.70e+02	4.70e+02	2.29e+00	4.59e+02	4.94e+02	4.67e+02	4.70e+02	9.34e+00	5.41e+02	6.13e+02	5.89e+02	5.86e+02	2.32e+01
28	3.00e+02		3.00e+02	3.00e+02	3.00e+02	0.00e+00	4.59e+02	4.59e+02	4.59e+02	4.59e+02	3.52e-12	4.78e+02	5.56e+02	4.99e+02	5.14e+02	3.09e+01
29	4.23e+02		4.76e+02	4.32e+02	4.42e+02	2.04e+01	4.07e+02	6.10e+02	4.92e+02	5.00e+02	4.13e+01	6.37e+02	1.09e+03	8.47e+02	8.10e+02	1.48e+02
30	1.96e+03		1.99e+03	1.99e+03	1.98e+03	1.18e+01	5.79e+05	5.82e+05	5.80e+05	5.80e+05	4.22e+02	2.09e+03	2.47e+03	2.21e+03	2.23e+03	1.08e+02

Moreover, our MLS-EDA considerably outperforms the basic EMNA_g on the CEC 2014 and CEC 2017 tests. Although our method has a slightly higher time cost than EMNA_g does, our proposed mechanisms have still been demonstrated to be effective in improving the performance of the basic EDA.

Moreover, the EDA usually requires a large population size to avoid stagnation [18]. However, our MLS-EDA can achieve promising performance with a relatively small population size, thus demonstrating the effectiveness of our modifications in enriching the population diversity.

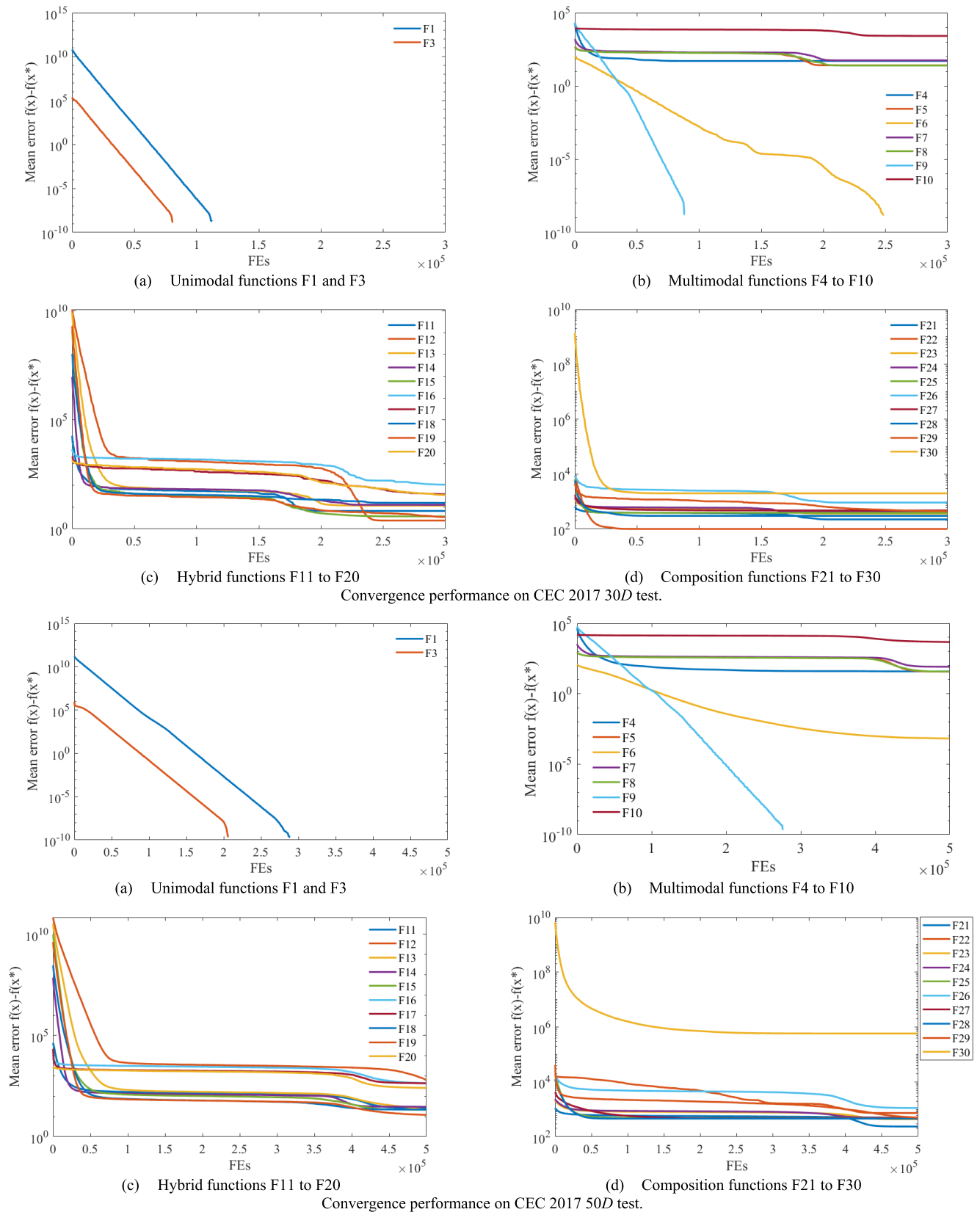


FIGURE 10. Convergence performance of the MLS-EDA on the CEC 2017 30D, 50D and 100D tests.

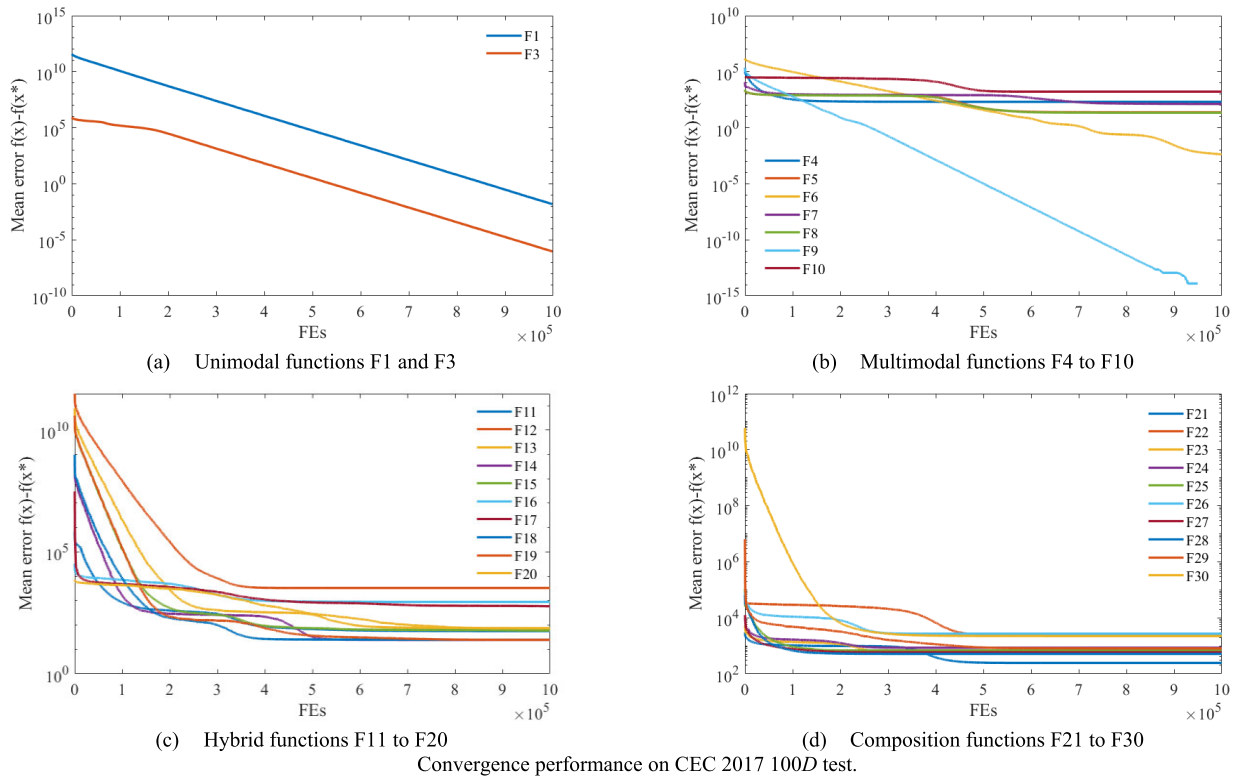


FIGURE 10. (Continued) Convergence performance of the MLS-EDA on the CEC 2017 30D, 50D and 100D tests.

TABLE 15. Mean errors obtained from seven algorithms on the CEC 2014 30D TEST.

No.	MLS-EDA	EMNA _g	RWGEDA	DOLTLBO	VCS	CPI-JADE	L-SHADE
	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}
01	0.00e+00 _{0.00e+00}	1.85e+08 _{2.56e+07}	0.00e+00 _{0.00e+00}	1.84e+05 _{1.43e+05}	4.39e+00 _{3.07e+01}	0.00e+00 _{0.00e+00}	0.00e+00 _{0.00e+00}
02	0.00e+00 _{0.00e+00}	2.00e+10 _{1.13e+09}	0.00e+00 _{0.00e+00}	3.12e+03 _{6.89e+03}	0.00e+00 _{0.00e+00}	0.00e+00 _{0.00e+00}	0.00e+00 _{0.00e+00}
03	0.00e+00 _{0.00e+00}	2.47e+04 _{2.26e+03}	0.00e+00 _{0.00e+00}	6.07e-01 _{9.62e-01}	1.62e-08 _{1.16e-07}	0.00e+00 _{0.00e+00}	0.00e+00 _{0.00e+00}
04	0.00e+00 _{0.00e+00}	2.43e+03 _{4.62e+02}	0.00e+00 _{0.00e+00}	8.58e+01 _{3.82e+01}	1.27e+00 _{9.10e+00}	0.00e+00 _{0.00e+00}	0.00e+00 _{0.00e+00}
05	2.00e+01 _{3.69e-02}	2.09e+01 _{4.40e-02}	2.00e+01 _{6.31e-04}	2.07e+01 _{1.24e-01}	2.04e+01 _{8.50e-02}	2.04e+01 _{4.81e-02}	2.01e+01 _{6.25e-02}
06	0.00e+00 _{0.00e+00}	5.49e+00 _{1.20e+00}	0.00e+00 _{0.00e+00}	2.30e+01 _{4.37e+00}	8.73e+00 _{3.38e+00}	3.41e+00 _{3.48e+00}	3.54e-05 _{2.53e-04}
07	0.00e+00 _{0.00e+00}	2.33e+02 _{4.66e+00}	0.00e+00 _{0.00e+00}	1.93e-01 _{5.06e-01}	3.38e-04 _{1.71e-03}	3.87e-04 _{1.99e-03}	0.00e+00 _{0.00e+00}
08	2.80e+01 _{8.28e+00}	4.33e+01 _{9.47e+00}	3.86e+01 _{1.04e+01}	9.57e+01 _{2.04e+01}	4.96e+01 _{5.66e+01}	0.00e+00 _{0.00e+00}	0.00e+00 _{0.00e+00}
09	2.34e+01 _{8.23e+00}	3.48e+01 _{1.08e+01}	3.66e+01 _{9.03e+00}	1.00e+02 _{2.24e+01}	1.36e+02 _{4.4e+01}	2.64e+01 _{3.48e+00}	8.19e+00 _{4.01e+00}
10	2.09e+03 _{8.08e+02}	8.91e+02 _{3.01e+02}	3.21e+03 _{8.19e+02}	3.11e+03 _{5.01e+02}	1.92e+02 _{8.09e+02}	5.76e-01 _{1.44e-01}	4.08e-03 _{1.02e-02}
11	2.38e+03 _{7.34e+02}	1.37e+03 _{1.99e+02}	3.34e+03 _{5.86e+02}	3.31e+03 _{5.95e+02}	1.97e+03 _{4.81e+02}	1.99e+03 _{2.38e+02}	1.20e+03 _{4.93e+02}
12	2.07e-02 _{5.74e-02}	2.43e+00 _{1.88e-01}	2.31e-02 _{1.90e-02}	9.79e-01 _{2.76e-01}	4.42e-01 _{1.75e-01}	3.73e-01 _{6.08e-02}	1.07e-01 _{3.69e-02}
13	2.09e-01 _{3.79e-02}	4.74e+00 _{1.17e-01}	2.32e-01 _{5.20e-02}	3.37e-01 _{8.46e-02}	2.98e-01 _{6.61e-02}	1.96e-01 _{3.05e-02}	5.01e-02 _{1.32e-02}
14	2.05e-01 _{4.13e-02}	1.06e+02 _{4.47e+00}	2.49e-01 _{3.67e-02}	2.37e-01 _{9.06e-02}	2.84e-01 _{7.26e-02}	2.20e-01 _{3.14e-02}	2.79e-01 _{2.93e-02}
15	3.05e+00 _{8.12e-01}	1.08e+02 _{6.65e+01}	3.31e+00 _{1.11e+00}	1.72e+01 _{7.76e+00}	3.29e+00 _{9.56e-01}	3.63e+00 _{4.33e-01}	2.89e+00 _{6.90e-01}
16	9.56e+00 _{8.45e-01}	1.27e+01 _{1.78e-01}	1.15e+01 _{9.94e-01}	1.14e+01 _{5.82e-01}	9.97e+00 _{6.84e-01}	9.68e+00 _{3.46e-01}	7.34e+00 _{9.63e-01}
17	2.39e+01 _{1.93e+01}	3.53e+06 _{2.07e+06}	9.21e+01 _{1.26e+02}	9.80e+03 _{1.16e+04}	1.07e+03 _{4.19e+02}	1.17e+03 _{3.25e+02}	2.07e+02 _{9.47e+01}
18	5.37e+00 _{2.37e+00}	2.04e+07 _{2.83e+07}	8.49e+00 _{2.55e+00}	6.21e+02 _{8.30e+02}	1.11e+03 _{2.56e+03}	8.98e+01 _{3.06e+01}	6.22e+00 _{2.52e+00}
19	1.38e+00 _{8.71e-01}	2.76e+01 _{1.27e+01}	1.69e+00 _{6.55e-01}	1.15e+01 _{1.22e+01}	1.11e+01 _{1.93e+01}	4.72e+00 _{7.07e-01}	2.63e+00 _{7.36e-01}
20	3.80e+00 _{1.67e+00}	4.46e+03 _{1.81e+03}	6.15e+00 _{2.57e+00}	2.50e+02 _{1.4e+02}	6.92e+01 _{3.89e+01}	1.46e+01 _{6.90e+00}	2.68e+00 _{1.38e+00}
21	1.04e+01 _{2.04e+01}	1.85e+05 _{1.56e+05}	9.45e+01 _{1.15e+02}	4.70e+03 _{4.29e+03}	9.05e+02 _{9.95e+02}	2.60e+02 _{1.04e+02}	7.30e+01 _{7.06e+01}
22	8.21e+01 _{6.22e+01}	3.26e+02 _{2.11e+02}	2.04e+02 _{1.28e+02}	3.30e+02 _{1.48e+02}	1.69e+02 _{1.18e+02}	1.14e+02 _{3.8e+01}	2.67e+01 _{2.41e+01}
23	2.00e+02 _{0.00e+00}	3.82e+02 _{1.84e+01}	2.00e+02 _{0.00e+00}	2.00e+02 _{0.00e+00}	2.00e+02 _{0.00e+00}	3.15e+02 _{2.00e+00}	3.15e+02 _{2.00e+00}
24	2.00e+02 _{3.81e+00}	2.32e+02 _{2.37e+00}	2.00e+02 _{2.34e+00}	2.00e+02 _{0.00e+00}	2.00e+02 _{0.00e+00}	2.25e+02 _{2.83e+00}	2.24e+02 _{2.23e+00}
25	2.00e+02 _{0.00e+00}	2.09e+02 _{4.91e-01}	2.00e+02 _{1.11e-07}	2.00e+02 _{0.00e+00}	2.00e+02 _{0.00e+00}	2.03e+02 _{4.92e-01}	2.03e+02 _{8.83e-02}
26	1.00e+02 _{3.89e-02}	1.07e+02 _{9.30e-01}	1.00e+02 _{6.00e-02}	1.32e+02 _{4.67e+01}	1.08e+02 _{2.71e+01}	1.02e+02 _{1.40e+01}	1.00e+02 _{1.14e-02}
27	2.00e+02 _{0.00e+00}	6.30e+02 _{4.50e+01}	2.37e+02 _{4.88e+01}	2.00e+02 _{0.00e+00}	2.00e+02 _{0.00e+00}	3.46e+02 _{5.01e+01}	3.00e+02 _{0.00e+00}
28	2.00e+02 _{6.79e-08}	1.18e+03 _{1.95e+02}	2.90e+02 _{2.07e+02}	2.00e+02 _{0.00e+00}	2.00e+02 _{0.00e+00}	8.13e+02 _{3.91e+01}	8.43e+02 _{1.34e+01}
29	1.18e+02 _{2.34e+01}	5.33e+03 _{2.15e+03}	1.17e+02 _{2.39e+01}	1.57e+03 _{6.68e+02}	9.03e+02 _{4.09e+02}	7.09e+02 _{5.00e+01}	7.16e+02 _{2.57e+00}
30	3.65e+02 _{2.84e+01}	7.97e+03 _{3.63e+03}	3.83e+02 _{4.44e+01}	2.38e+03 _{6.46e+02}	1.83e+03 _{8.49e+02}	1.21e+03 _{5.39e+02}	1.29e+03 _{4.70e+02}

TABLE 16. Mean errors obtained from seven algorithms on the CEC 2014 50D test.

No.	MLS-EDA	EMNA _g	RWGEDA	DOLTLBO	VCS	CPI-JADE	L-SHADE
	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}
01	0.00e+00	1.38e+09	0.00e+00	1.35e+03	4.72e+02	1.21e+03	1.29e+03
02	0.00e+00	6.52e+10	0.00e+00	2.30e+08	0.00e+00	0.00e+00	0.00e+00
03	0.00e+00	5.60e+04	0.00e+00	4.72e+02	0.00e+00	0.00e+00	0.00e+00
04	5.59e-01	1.11e+04	1.79e+01	1.52e+02	1.73e+01	1.78e+01	2.12e+01
05	2.00e+01	2.11e+01	2.00e+01	2.10e+01	2.06e+01	2.04e+01	2.02e+01
06	3.76e-05	2.27e+01	2.25e-02	4.63e+01	2.01e+01	6.39e+00	5.69e+01
07	0.00e+00	5.81e+02	0.00e+00	1.55e+01	1.23e-03	0.00e+00	0.00e+00
08	3.71e+01	1.87e+02	8.01e+01	2.19e+02	7.63e+01	0.00e+00	0.00e+00
09	3.63e+01	1.84e+02	5.72e+01	2.30e+02	2.10e+02	5.54e+01	9.62e+02
10	5.05e+03	4.63e+03	5.25e+03	5.93e+03	8.86e+01	2.97e+00	4.51e-02
11	4.38e+03	5.00e+03	5.60e+03	6.69e+03	4.57e+03	4.60e+03	3.71e+03
12	2.36e-02	2.98e+00	5.88e-03	1.98e+00	5.11e-01	3.73e-01	1.14e-01
13	3.42e-01	5.16e+00	2.98e-01	5.41e-01	4.88e-01	3.07e-01	6.93e-02
14	2.51e-01	1.35e+02	2.73e-01	3.00e-01	3.06e-01	3.19e-01	3.62e-01
15	4.77e+00	4.10e+04	6.12e+00	1.01e+02	6.40e+00	8.12e+00	5.45e+00
16	1.94e+01	2.23e+01	2.03e+01	1.98e+01	1.90e+01	1.81e+01	1.49e+01
17	2.50e+02	1.38e+08	7.29e+02	9.95e+04	1.21e+03	2.40e+03	1.27e+03
18	6.08e+02	4.49e+09	1.66e+01	7.25e+02	1.44e+03	1.85e+02	9.76e+01
19	8.17e+00	3.11e+02	7.78e+00	4.27e+01	2.25e+01	1.24e+01	6.97e+00
20	7.92e+02	3.91e+03	1.09e+01	5.21e+02	1.86e+02	2.28e+02	1.33e+01
21	2.76e+02	1.35e+05	3.98e+02	1.01e+05	1.08e+03	1.39e+02	4.81e+02
22	2.03e+02	3.55e+03	5.89e+02	1.01e+03	7.73e+02	2.95e+02	6.66e+01
23	2.00e+02	5.60e+02	2.00e+02	2.00e+02	2.00e+02	3.44e+02	3.44e+02
24	2.00e+02	2.74e+02	2.00e+02	2.00e+02	2.00e+02	2.75e+02	2.75e+02
25	2.05e+02	2.18e+02	2.03e+02	2.00e+02	2.00e+02	2.10e+02	2.05e+02
26	1.00e+02	1.99e+02	1.00e+02	1.83e+02	1.17e+02	1.00e+02	1.00e+02
27	3.00e+02	1.27e+03	3.00e+02	2.00e+02	2.00e+02	4.58e+02	3.18e+02
28	8.93e+02	2.73e+03	1.00e+03	2.00e+02	2.00e+02	1.23e+03	1.10e+03
29	3.39e+02	2.88e+05	3.49e+02	5.26e+02	9.51e+02	8.36e+02	8.44e+02
30	8.61e+03	7.85e+05	8.71e+03	1.87e+04	7.74e+03	1.09e+04	8.95e+03

TABLE 17. Mean errors obtained from seven algorithms on the CEC 2014 100D test.

No.	MLS-EDA	EMNA _g	RWGEDA	DOLTLBO	VCS	CPI-JADE	L-SHADE
	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}	Mean Error _{SD}
01	3.20e+02	1.75e+09	1.62e+04	4.52e+07	4.99e+05	2.03e+04	1.84e+05
02	2.77e-03	1.59e+11	1.92e+04	4.48e+09	6.70e+00	6.11e-08	0.00e+00
03	1.09e-06	1.78e-07	3.55e-03	2.11e+04	5.14e+01	0.00e+00	0.00e+00
04	1.52e+02	3.20e+04	1.53e+02	5.41e+02	1.56e+02	8.96e+00	1.69e+02
05	2.00e+01	2.13e+01	2.13e+01	2.13e+01	2.10e+01	2.05e+01	2.05e+01
06	6.69e+00	7.98e+01	1.23e+02	1.26e+02	6.78e+01	3.26e+01	8.82e+00
07	0.00e+00	1.74e+03	4.92e-03	2.14e+01	0.00e+00	8.22e-04	0.00e+00
08	7.84e+02	7.12e+02	9.45e+02	5.49e+02	3.34e+02	0.00e+00	1.53e-03
09	7.44e+02	7.27e+02	9.37e+02	6.46e+02	6.70e+02	1.90e+02	1.77e+01
10	1.25e+04	1.81e+04	2.97e+04	1.56e+04	3.37e+03	1.60e+01	1.67e+01
11	1.24e+04	1.62e+04	2.98e+04	1.38e+04	1.17e+04	1.18e+04	1.09e+04
12	1.91e-02	1.19e+00	4.02e+00	2.16e+00	1.47e+00	5.00e-01	1.94e-01
13	4.70e-01	6.99e+00	6.60e-01	5.86e-01	5.57e-01	4.34e-01	1.36e-01
14	2.78e-01	5.09e+02	2.89e-01	2.81e-01	3.94e-01	3.17e-01	3.72e-01
15	6.38e+01	9.97e+05	7.88e+01	3.36e+02	1.35e+01	2.84e+01	1.23e+01
16	4.35e+01	4.64e+01	4.64e+01	4.41e+01	4.27e+01	4.03e+01	3.75e+01
17	6.70e+03	3.36e+08	8.92e+03	1.22e+06	9.24e+04	4.90e+03	4.42e+03
18	2.70e+02	1.14e+10	3.79e+02	1.32e+05	2.79e+03	3.84e+02	2.25e+02
19	9.23e+01	1.81e+03	9.47e+01	1.55e+02	1.07e+02	1.04e+02	9.65e+01
20	1.82e+02	7.56e+04	2.75e+02	9.83e+03	3.83e+03	5.64e+02	1.52e+02
21	3.85e+03	3.00e+07	5.80e+03	5.40e+05	1.21e+04	2.68e+03	1.91e+03
22	1.75e+03	1.75e+03	4.03e+03	2.56e+03	1.56e+03	1.35e+03	7.21e+02
23	3.48e+02	1.02e+03	3.48e+02	2.00e+02	2.00e+02	3.48e+02	3.48e+02
24	3.82e+02	4.07e+02	3.68e+02	2.00e+02	2.00e+02	3.98e+02	3.93e+02
25	2.15e+02	2.39e+02	2.23e+02	2.00e+02	2.00e+02	2.54e+02	2.00e+02
26	1.76e+02	2.00e+02	2.00e+02	2.00e+02	1.89e+02	2.00e+02	2.00e+02
27	4.57e+02	3.70e+03	1.15e+03	2.00e+02	2.00e+02	1.08e+03	3.90e+02
28	3.12e+03	1.09e+04	8.32e+03	2.00e+02	2.00e+02	2.63e+03	2.32e+03
29	7.12e+02	5.68e+07	7.41e+02	2.00e+02	2.00e+02	9.14e+02	8.02e+02
30	6.48e+03	3.27e+06	8.69e+03	1.07e+05	4.32e+04	8.47e+03	8.69e+03

TABLE 22. Comparison of the computational efficiencies of the eight algorithms on the CEC 2017 test suite.

No.	MLS-EDA			EMNA _g			NRO			MRDE			RWGEDA			ELSHADE-SPACMA			ACoS-JADE			HSES		
	30D	50D	100D	30D	50D	100D	30D	50D	100D	30D	50D	100D	30D	50D	100D	30D	50D	100D	30D	50D	100D	30D	50D	100D
	(unit: second)																							
01	1.96	4.69	39.7	1.73	4.26	25.1	5.17	11.10	35.8	7.71	14.73	49.3	1.99	4.56	32.5	2.03	5.03	33.6	4.12	9.28	42.9	2.40	8.37	69.2
03	1.94	4.77	40.0	1.75	4.23	25.1	5.11	10.88	36.4	7.87	14.83	50.0	2.08	4.57	32.3	2.03	4.97	33.5	4.27	9.79	43.1	2.35	7.69	68.6
04	2.05	4.64	40.7	1.75	4.12	25.1	5.10	11.10	36.1	7.99	15.15	50.2	2.04	4.74	32.4	1.92	4.74	32.8	6.49	12.86	50.6	2.35	7.64	69.4
05	2.56	5.60	43.5	2.13	5.10	29.1	5.36	12.19	40.1	8.62	16.64	55.4	2.32	5.68	35.9	2.47	5.89	37.2	4.58	9.66	44.3	2.68	8.93	73.6
06	3.49	8.31	55.6	3.18	8.03	40.6	6.00	15.49	51.9	9.21	18.47	64.8	3.34	8.80	48.0	3.43	8.91	49.3	6.28	14.93	62.7	3.75	11.83	89.3
07	2.40	5.75	43.8	2.25	5.21	29.2	5.39	12.12	40.1	8.24	15.75	53.4	2.36	5.90	36.0	2.38	5.85	37.2	4.74	9.60	44.3	2.83	8.86	73.9
08	2.42	5.78	44.1	2.24	5.23	29.7	5.37	12.29	40.9	8.55	17.03	55.9	2.40	5.95	36.3	2.41	6.03	37.1	4.76	9.80	45.2	2.81	8.87	74.4
09	2.32	5.78	43.9	2.43	5.11	29.4	5.43	12.30	40.3	8.20	16.09	53.8	2.41	5.73	35.7	2.44	5.97	37.2	6.91	14.06	57.3	2.85	9.01	74.5
10	2.62	6.39	46.6	2.41	5.75	32.0	5.68	13.34	44.0	10.79	20.79	66.6	2.61	6.56	39.2	2.66	6.73	39.5	5.41	10.30	47.7	3.00	9.41	77.3
11	2.22	5.10	41.6	1.98	4.54	27.0	5.36	11.68	38.3	8.12	15.72	52.0	2.17	5.38	33.5	2.06	5.33	34.3	5.32	9.84	51.7	2.59	8.28	71.1
12	2.46	5.64	43.7	2.18	5.16	29.5	5.43	12.40	41.0	8.58	16.54	56.7	2.31	5.89	36.0	2.37	5.93	36.6	5.85	11.92	54.4	2.77	8.83	74.4
13	2.26	5.13	41.5	2.02	4.62	27.4	5.38	11.84	38.8	8.41	15.81	51.9	2.17	5.36	33.9	2.26	5.41	34.6	5.28	10.05	51.7	2.67	8.23	71.8
14	7.17	6.34	47.0	2.42	5.74	32.3	5.56	13.16	43.5	9.46	18.90	60.6	4.66	6.46	39.1	2.42	6.45	39.7	5.23	10.80	53.1	3.00	9.35	77.7
15	2.22	5.02	41.3	1.91	4.46	26.3	5.23	11.46	38.7	8.21	15.73	52.2	2.13	5.26	33.3	2.11	5.26	33.9	5.27	9.79	46.5	2.53	8.00	70.9
16	2.31	5.48	42.9	2.10	4.88	28.3	5.40	12.34	40.5	8.97	17.47	59.6	2.23	5.59	35.6	2.27	5.69	34.6	4.88	9.84	44.7	2.73	8.81	72.8
17	3.24	7.70	51.9	3.06	7.25	37.5	6.04	15.33	50.5	10.33	20.50	69.2	3.11	7.73	44.7	3.22	8.11	44.0	5.70	12.06	53.8	3.65	11.33	86.6
18	2.48	5.37	43.0	2.14	4.88	28.3	5.39	12.12	40.3	8.71	17.04	57.8	2.24	5.54	35.0	2.17	5.67	35.8	5.31	9.99	51.7	2.84	8.73	72.7
19	7.68	20.36	101.5	7.54	19.72	86.8	9.09	30.19	101.1	14.47	32.02	115.9	7.58	20.00	93.2	7.74	20.84	95.0	10.42	25.06	109.6	8.45	23.95	168.4
20	3.46	8.41	53.8	3.16	7.75	39.0	6.05	15.81	52.1	10.18	21.31	71.9	5.79	8.13	46.3	3.29	8.58	47.0	6.04	12.93	56.7	3.89	11.46	89.6
21	3.82	11.46	90.1	3.75	10.99	75.9	6.08	17.27	63.8	9.99	22.29	102.2	3.82	11.34	81.5	3.70	11.54	80.5	6.06	15.41	90.9	4.43	14.70	139.2
22	4.11	12.56	94.9	4.15	12.07	80.3	6.40	18.70	68.9	9.99	26.00	113.7	4.28	12.82	86.5	4.26	12.95	86.8	8.55	16.68	95.4	4.77	15.46	143.8
23	4.61	14.18	114.2	4.56	14.04	98.9	6.56	20.05	76.6	11.05	25.18	128.4	4.67	14.69	104.2	4.58	14.52	105.0	7.17	18.44	114.2	5.21	16.91	169.2
24	4.98	15.28	118.6	4.97	15.06	103.7	6.66	21.29	80.5	11.21	25.90	130.0	4.98	15.81	109.0	4.95	16.15	108.0	7.38	19.58	118.7	5.55	18.47	173.6
25	4.34	14.50	127.8	4.41	14.23	111.7	6.35	19.60	80.3	10.44	24.81	135.3	4.42	14.89	117.1	4.37	14.79	116.8	8.15	22.70	134.4	5.06	17.61	184.9
26	5.57	17.31	137.0	5.41	17.20	124.1	7.00	22.95	89.3	11.53	27.85	148.3	5.39	17.63	129.1	5.45	17.66	128.3	8.11	21.60	139.0	6.16	20.50	199.1
27	6.04	19.72	159.6	6.04	19.64	145.6	7.19	24.99	99.6	12.78	31.00	172.6	5.95	19.62	150.1	5.97	20.64	151.2	8.93	26.83	168.2	6.79	23.44	225.4
28	5.22	17.85	150.5	5.32	17.56	136.5	6.77	22.47	90.4	11.27	27.84	161.7	5.27	17.19	141.1	5.35	17.91	145.1	9.79	25.87	161.5	6.05	21.28	213.5
29	4.51	12.91	96.5	4.43	12.56	82.9	6.75	19.82	71.9	12.15	25.22	112.8	4.58	12.82	88.2	4.48	13.31	89.3	7.00	17.18	97.6	5.12	16.27	147.8
30	8.89	24.93	146.8	8.98	25.07	133.2	10.28	34.52	121.4	16.19	37.98	160.1	8.79	25.17	136.1	8.98	25.77	139.1	12.06	31.49	150.6	9.69	29.33	212.0

Two possibilities should be addressed in future work. First, the MLS mechanism used in the MLS-EDA is an efficient local exploration method that can also be employed to enhance the performance of other algorithms. Second, although our MLS-EDA has shown promising performance in addressing small- and medium-scale problems in this study, the time complexity of MLS-EDA, which is related to the problem dimensionality, will limit its applicability in high-dimensional cases. Developing a more efficient EDA framework with a lower time complexity will require further investigation.

APPENDIXES

APPENDIX A

See Table 13.

APPENDIX B

See Figure 9.

APPENDIX C

See Table 14.

APPENDIX D

See Figure 10.

APPENDIX E

EXPERIMENTAL RESULTS ACHIEVED BY THE MLS-EDA, EMNA_g, RWGEDA, DOLTLBO, VCS, CPI-JADE AND L-SHADE ON THE CEC 2014 TEST SUITE

See Tables 15–17.

APPENDIX F

COMPARISON OF THE COMPUTATIONAL COSTS OF THE SEVEN METHODS ON THE CEC 2014 TEST SUITE

See Table 18.

APPENDIX G

EXPERIMENTAL RESULTS ACHIEVED BY THE MLS-EDA, EMNA_g, NRO, MRDE, RWGEDA, ELSHADE-SPACMA, ACOS-JADE AND HSES ON THE CEC 2017 TEST SUITE

See Tables 19–21.

APPENDIX H

COMPARISON OF THE COMPUTATIONAL COSTS OF THE EIGHT METHODS ON THE CEC 2017 TEST SUITE

See Table 22.

REFERENCES

- [1] J. R. Sampson, "Adaptation in natural and artificial systems (John H. Holland)," *SIAM Rev.*, vol. 18, no. 3, pp. 529–530, Jul. 1976.
- [2] M. Kubat, "Neural networks: A comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7," *Knowl. Eng. Rev.*, vol. 13, no. 4, pp. 409–412, Apr. 2001.
- [3] L. Liu, S. Yang, and D. Wang, "Particle swarm optimization with composite particles in dynamic environments," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1634–1648, Dec. 2010.
- [4] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution*. Berlin, Germany: Springer-Verlag, 2005.
- [5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [6] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.
- [7] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011.
- [8] M. D. Li, H. Zhao, X. W. Weng, and T. Han, "A novel nature-inspired algorithm for optimization: Virus colony search," *Adv. Eng. Softw.*, vol. 92, pp. 65–88, Feb. 2016.
- [9] A. A. Heidari, S. Mirjalili, H. Farris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
- [10] Z. Wei, C. Huang, X. Wang, T. Han, and Y. Li, "Nuclear reaction optimization: A novel and powerful physics-based algorithm for global optimization," *IEEE Access*, vol. 7, pp. 66084–66109, 2019.
- [11] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1658–1665.

- [12] Y. Wang, Z.-Z. Liu, J. Li, H.-X. Li, and G. G. Yen, "Utilizing cumulative population distribution information in differential evolution," *Appl. Soft Comput.*, vol. 48, pp. 329–346, Nov. 2016.
- [13] Y. Xu, Z. Yang, X. Li, H. Kang, and X. Yang, "Dynamic opposite learning enhanced teaching-learning-based optimization," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 104966.
- [14] X. Wang, H. Zhao, T. Han, H. Zhou, and C. Li, "A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem," *Appl. Soft Comput.*, vol. 78, pp. 240–260, May 2019.
- [15] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms*, vol. 2. Boston, MA, USA: Springer, 2002.
- [16] P. Larranaga, R. Etxeberria, J. A. Lozano, and J. M. Pena, "Combinatorial optimization by learning and simulation of Bayesian and Gaussian networks," in *Proc. 16th Conf. Uncertain. Artif. Intell.*, Nov. 2000, pp. 343–352.
- [17] M. Pelikan and H. Muehlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing*. London, U.K.: Springer, 1999, pp. 521–535.
- [18] Z. Ren, Y. Liang, L. Wang, A. Zhang, B. Pang, and B. Li, "Anisotropic adaptive variance scaling for Gaussian estimation of distribution algorithm," *Knowl.-Based Syst.*, vol. 146, pp. 142–151, Apr. 2018.
- [19] J. Grahl, P. A. N. Bosman, and F. Rothlauf, "The correlation-triggered adaptive variance scaling IDEA," in *Proc. 8th Annu. Conf. Genetic Evol. Comput. (GECCO)*, 2006, p. 397.
- [20] P. A. N. Bosman and D. Thierens, "Adaptive variance scaling in continuous multi-objective estimation-of-distribution algorithms," in *Proc. 9th Annu. Conf. Genetic Evol. Comput. (GECCO)*, 2007, p. 500.
- [21] P. A. N. Bosman, J. Grahl, and D. Thierens, "Benchmarking parameter-free AMaLGaM on functions with and without noise," *Evol. Comput.*, vol. 21, no. 3, pp. 445–469, Sep. 2013.
- [22] P. A. N. Bosman, J. Grahl, and D. Thierens, "Enhancing the performance of maximum-likelihood Gaussian EDAs using anticipated mean shift," in *Proc. Int. Conf. Parallel Problem Solving Nature*, in Lecture Notes in Computer Science, 2008, pp. 133–143.
- [23] Z. Ren, C. He, D. Zhong, S. Huang, and Y. Liang, "Enhance continuous estimation of distribution algorithm by variance enlargement and reflecting sampling," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 3441–3447.
- [24] Y. Liang, Z. Ren, X. Yao, Z. Feng, A. Chen, and W. Guo, "Enhancing Gaussian estimation of distribution algorithm by exploiting evolution direction with archive," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 140–152, Jan. 2020.
- [25] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, Jun. 2001.
- [26] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, Sep. 2005, pp. 1769–1776.
- [27] I. Loshchilov, "CMA-ES with restarts for solving CEC 2013 benchmark problems," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 369–376.
- [28] N. Hansen, "Benchmarking a BI-population CMA-ES on the BBOB-2009 noisy testbed," in *Proc. 11th Annu. Conf. Companion Genetic Evol. Comput. Conf. (GECCO)*, 2009, p. 2397.
- [29] G. A. Jastrebski and D. V. Arnold, "Improving evolution strategies through active covariance matrix adaptation," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 2814–2821.
- [30] I. Loshchilov, M. Schoenauer, and M. Sebag, "Alternative restart strategies for CMA-ES," in *Proc. Int. Conf. Parallel Problem Solving Nature*, in Lecture Notes in Computer Science, 2012, pp. 296–305.
- [31] B. Yuan and M. Gallagher, "On the importance of diversity maintenance in estimation of distribution algorithms," in *Proc. Conf. Genetic Evol. Comput. (GECCO)*, 2005, p. 719.
- [32] X. Huang, P. Jia, and B. Liu, "Controlling chaos by an improved estimation of distribution algorithm," *Math. Comput. Appl.*, vol. 15, no. 5, pp. 866–871, Dec. 2010.
- [33] T. Miquélez, E. Bengoetxea, A. Mendiburu, and P. Larrañaga, "Combining Bayesian classifiers and estimation of distribution algorithms for optimization in continuous domains," *Connection Sci.*, vol. 19, no. 4, pp. 297–319, Dec. 2007.
- [34] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Regularized continuous estimation of distribution algorithms," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2412–2432, May 2013.
- [35] B. Qian, Z.-C. Li, and R. Hu, "A copula-based hybrid estimation of distribution algorithm for m-machine reentrant permutation flow-shop scheduling problem," *Appl. Soft Comput.*, vol. 61, pp. 921–934, Dec. 2017.
- [36] L. PourMohammadBagher, M. M. Ebadzadeh, and R. Safabakhsh, "Graphical model based continuous estimation of distribution algorithm," *Appl. Soft Comput.*, vol. 58, pp. 388–400, Sep. 2017.
- [37] X. Wang, H. Zhao, T. Han, Z. Wei, Y. Liang, and Y. Li, "A Gaussian estimation of distribution algorithm with random walk strategies and its application in optimal missile guidance handover for multi-UCAV in over-the-horizon air combat," *IEEE Access*, vol. 7, pp. 43298–43317, 2019.
- [38] P. Yang, K. Tang, and X. Lu, "Improving estimation of distribution algorithm on multimodal problems by detecting promising areas," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1438–1449, Aug. 2015.
- [39] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, "Seeking multiple solutions: An updated survey on niching methods and their applications," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 518–538, Aug. 2017.
- [40] Q. Yang, W.-N. Chen, Y. Li, C. L. P. Chen, X.-M. Xu, and J. Zhang, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 636–650, Mar. 2017.
- [41] X. Qi, K. Li, and W. D. Potter, "Estimation of distribution algorithm enhanced particle swarm optimization for water distribution network optimization," *Frontiers Environ. Sci. Eng.*, vol. 10, no. 2, pp. 341–351, Feb. 2015.
- [42] X. Song and L. Tang, "A novel hybrid differential evolution-estimation of distribution algorithm for dynamic optimization problem," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 1710–1717.
- [43] F. Zhao, Z. Shao, J. Wang, and C. Zhang, "A hybrid differential evolution and estimation of distribution algorithm based on neighbourhood search for job shop scheduling problems," *Int. J. Prod. Res.*, vol. 54, no. 4, pp. 1039–1060, May 2015.
- [44] F. Zhao, Z. Shao, J. Wang, and C. Zhang, "A hybrid optimization algorithm based on chaotic differential evolution and estimation of distribution," *Comput. Appl. Math.*, vol. 36, no. 1, pp. 433–458, May 2015.
- [45] Z. Sun and X. Gu, "Hybrid algorithm based on an estimation of distribution algorithm and cuckoo search for the no idle permutation flow shop scheduling problem with the total tardiness criterion minimization," *Sustainability*, vol. 9, no. 6, p. 953, Jun. 2017.
- [46] S. Elsayed, N. Hamza, and R. Sarker, "Testing united multi-operator evolutionary algorithms-II on single objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 2966–2973.
- [47] A. Kumar, R. K. Misra, and D. Singh, "Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1835–1842.
- [48] G. Zhang and Y. Shi, "Hybrid sampling evolution strategy for solving single objective bound constrained problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–7.
- [49] J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," Zhengzhou Univ., Zhengzhou, China, and Nanyang Technol. Univ., Singapore, Tech. Rep., 2013.
- [50] 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," Nanyang Technol. Univ., Jordan Univ. Sci. Technol., Zhengzhou Univ., Zhengzhou, China, Tech. Rep., 2016.
- [51] N. Veček, M. Mernik, and M. Črepinšek, "A chess rating system for evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms," *Inf. Sci.*, vol. 277, pp. 656–679, Sep. 2014.
- [52] L. Gui, X. Xue, F. Yu, H. Wu, H. Wu, B. Wei, Y. Zhang, X. Li, and G. He, "A multi-role based differential evolution," *Swarm. Evol. Comput.*, vol. 50, Nov. 2019, Art. no. 100508.
- [53] A. A. Hadi, A. Wagdy, and K. Jambi, "Single-objective real-parameter optimization: Enhanced LSHADE-SPACMA algorithm," King Abdulaziz Univ., Jeddah, Saudi Arabia, Tech. Rep., 2018, doi: 10.13140/RG.2.2.33283.20005.
- [54] Z.-Z. Liu, Y. Wang, S. Yang, and K. Tang, "An adaptive framework to tune the coordinate systems in nature-inspired optimization algorithms," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1403–1416, Apr. 2019.



XIAOFEI WANG (Member, IEEE) was born in 1990. He received the M.S. degree in weapon science and technology from Air Force Engineering University, in 2015. He is currently pursuing the Ph.D. degree with Air Force Engineering University. His research interests are evolutionary algorithms and UCAV air combat decision.



HUI ZHAO was born in 1973. He received the M.S. degree from Air Force Engineering University, in 2002, and the Ph.D. degree in weapon science and technology from Air Force Engineering University, in 2011. He is currently a Doctor and a Professor/Doctoral Supervisor with the Department of Aeronautics and Astronautics Engineering, Air Force Engineering University. He has been engaged in teaching and researching on weapon systems and application engineering and optimal control.

...



TONG HAN was born in 1980. He received the M.S. degree from Air Force Engineering University, in 2005, and the Ph.D. degree in weapon science and technology from Air Force Engineering University, in 2008. He is currently an Associate Professor with the Department of Aeronautics and Astronautics Engineering, Air Force Engineering University. He has been engaged in teaching and researching on weapon systems and application engineering and multiobjective optimization.