An Estimation of Distribution Algorithm With Multi-Leader Search

XIAOFEI WANG (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 new, 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
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 Miqueléz’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 UMOEAI [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 EL-SHADE-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:

\[ G(\mu, C) = \frac{(2\pi)^{-n/2}}{|C|} \exp \left( -\frac{(X - \mu)^T (C)^{-1} (X - \mu)}{2} \right) \]  
(1)

where

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

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

\[ x_i = \mu + y, \quad y_i \sim N(0, C) \]  
(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 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 becomes 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) a tangent 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].

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

where

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

In (6), the \( \omega_i \) 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

\[
P_{\text{rank}}(i) = 1 + \left( 1 - (f(x_i))_{\text{rank}} \right) / NP,
\]

\( (f(x_i))_{\text{rank}} \in [1, 2, \ldots, NP] \) (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:

\[
\begin{align*}
\mu^{(E)}_j & = B^T \mu_j \\
x^{(E)}_j & = B^T x_j
\end{align*}
\]

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

\[
C = (BD)(BD)^T
\]

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:

\[
\mu^{(E)}_j = \mu^{(E)}_j + r \cdot (x^{(E)}_{i,j} - \mu^{(E)}_j), \quad r \sim U(0, 1)
\]

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

\[
x^{(E)}_{i,j} = \mu^{(E)}_j + D y_{i,j}, \quad y_{i,j} \sim N(0, 1)
\]

\( D \) in (11) is a vector of the square root of the eigenvalues, i.e., \( D = (\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_{\text{dim}}}) \), 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 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).

\[
\mu^{(E)}_j = \mu^{(E)}_j + r \cdot (\mu^{(E)}_j - x^{(E)}_{i,j}), \quad r \sim U(0, 1)
\]

(12)

\[
x^{(E)}_{i,j} = \mu^{(E)}_j + D y_{i,j}, \quad y_{i,j} \sim N(0, 1)
\]

(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 \( S \)) with an adaptive size is designed and employed to store the current dominant solutions. Initially, the size of \( S \) is equal to 1, meaning that
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).

$$x_i^{(E)} = x_i^{(E)} + \beta (x_{\text{leader}}^{(E)} - x_i^{(E)}) + r_1 \cdot x_{\text{leader}}^{(E)} - r_2 \cdot x_i^{(E)},$$

$$y_j \sim N(0, I), \quad r_1, r_2 \sim N(0, 1), \quad x_{\text{leader}} \in S$$

In (14), $x_{\text{leader}}$ 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:

$$x_i^{(E)} = x_i^{(E)} + \beta (|z_i| \cdot (x_i^{(E)} - x_{\text{leader}}^{(E)}) + D'y_i, \quad z_i \sim N(0, 1), \quad y_j \sim N(0, I), \quad x_{\text{leader}} \in S$$

where

$$D' = D \cdot (1 - FE_{\text{avg}}/FE_{\text{max}})$$

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 distribution mean point, thus enhancing the distribution and even the diversity of the population. As presented in (16), the elements of vector $D$ 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:

$$x_i = B \cdot x_i^{(E)}$$

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 ($FE_{max}$) to ensure a fair evaluation. The value of $FE_{max}$ 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 \( FEs_{\text{max}} \);
2: Produce the initial population \( X \) and evaluate, \( FEs = FEs + NP \);
3: Update the best solution \( x_{\text{Best}} \) in the present generation;
4: While \( FEs \leq FEs_{\text{max}} \), output \( f(x_{\text{best}}) \) 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: 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);
7: Calculate the eigenvalue decomposition of \( C \) using (9);
8: For each solution \( x \),
   \begin{align*}
   & \text{Calculate} \ P_{\text{rank}}(i); \\
   & \quad \text{Rotate} \ x_i \text{ into the eigen coordinate system and determine} \ x_i^{(E)} \text{ using (8)}; \\
   & \quad \text{If the algorithm falls into stagnation} \\
   & \quad \quad \text{Update the size of archive} \ S; \\
   & \quad \quad \text{If} \ P_{\text{rank}}(i) < 1 - |S|/NP \\
   & \quad \quad \quad \text{Update} \ x_i^{(E)} \text{ using the MLS behavior as expressed in (14)}; \\
   & \quad \quad \text{Else} \\
   & \quad \quad \quad \text{Update} \ x_i^{(E)} \text{ using the MLS behavior as expressed in (15) and (16)}; \\
   & \quad \text{End if} \\
   & \quad \text{Otherwise} \\
   & \quad \quad \text{If} \ P_{\text{rank}}(i) > 0.5 \\
   & \quad \quad \quad \text{Update} \ x_i^{(E)} \text{ using the enhanced mean as expressed in (10) and (11)}; \\
   & \quad \quad \text{Else} \\
   & \quad \quad \quad \text{Update} \ x_i^{(E)} \text{ using the enhanced mean as expressed in (12) and (13)}; \\
   & \quad \text{End if} \\
   & \quad \text{End if} \\
   & \quad \text{Rotate candidate} \ x_i^{(E)} \text{ into the normal coordinate system by (17)}; \\
   & \quad \text{Calculate the fitness} \ f(x_i); \\
   & \text{End for} \\
9: \text{ } FEs = FEs + NP \text{ 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 10D and the maximum size of set \( S \) set to 3D. 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 \( |S|_{\text{max}} \) 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 5 times with \( FEs_{\text{max}} = 300,000 \) 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, \( NP = 6D, 8D, 10D, 12D \) and 14D, 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 \( NP = 10D \). 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 \( O(D^2 \times \text{max}(D, NP/2)) \).

In Table 2, the rankings are assessed for settings of \( |S|_{\text{max}} = 1, D, 2D, 3D, 4D \) and 5D. When the value of \( |S|_{\text{max}} \) 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 \( |S|_{\text{max}} = 3 \cdot D \) yields the most promising results, as indicated by its highest ranking.
Overall, the optimal parameter settings of the MLS-EDA are $NP = 10 \cdot D$ and $|S|_{\text{max}} = 3 \cdot D$. 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 $FEs_{\text{max}}$ 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 MLS-EDA with different maximum sizes of the set $S$ on the CEC 2014 test suite for 30D problems.

| No. | $|S|_{\text{max}}=1$ | $|S|_{\text{max}}=2$ | $|S|_{\text{max}}=3$ | $|S|_{\text{max}}=4$ | $|S|_{\text{max}}=5$ |
|-----|-----------------|-----------------|-----------------|-----------------|-----------------|
|     | Mean error value | Mean error value | Mean error value | Mean error value | Mean error value |
| 01  | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          |
| 02  | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          |
| 03  | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          |
| 04  | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          |
| 05  | 2.00±01          | 2.00±01          | 2.00±01          | 2.00±01          | 2.00±01          |
| 06  | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          |
| 07  | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          | 0.00±00          |
| 08  | 2.95±01          | 2.89±01          | 2.83±01          | 2.74±01          | 2.69±01          |
| 09  | 2.98±01          | 2.93±01          | 2.88±01          | 2.82±01          | 2.77±01          |
| 10  | 2.09±03          | 2.04±03          | 2.15±03          | 2.09±03          | 2.05±03          |
| 11  | 2.45±03          | 1.98±03          | 2.40±03          | 2.38±03          | 2.38±03          |
| 12  | 1.99±02          | 3.97±02          | 2.17±02          | 2.07±02          | 2.10±02          |
| 13  | 2.15±01          | 2.07±01          | 2.03±01          | 2.09±01          | 2.06±01          |
| 14  | 1.97±01          | 2.34±01          | 2.23±01          | 2.05±01          | 2.26±01          |
| 15  | 2.85±00          | 2.60±00          | 2.79±00          | 3.05±00          | 2.44±00          |
| 16  | 9.59±00          | 9.23±00          | 9.60±00          | 9.56±00          | 9.39±00          |
| 17  | 2.77±01          | 3.15±01          | 2.25±01          | 2.39±01          | 4.01±01          |
| 18  | 6.88±00          | 6.50±00          | 5.54±00          | 5.37±00          | 5.35±00          |
| 19  | 7.07±01          | 1.19±00          | 1.23±00          | 1.38±00          | 1.71±00          |
| 20  | 4.92±00          | 4.24±00          | 4.28±00          | 3.80±00          | 4.74±00          |
| 21  | 1.13±01          | 6.82±00          | 6.23±00          | 1.04±01          | 1.05±01          |
| 22  | 1.13±02          | 9.61±01          | 1.04±02          | 8.21±01          | 8.55±01          |
| 23  | 2.00±02          | 2.00±02          | 2.00±02          | 2.00±02          | 2.00±02          |
| 24  | 2.02±02          | 2.01±02          | 2.01±02          | 2.00±02          | 2.01±02          |
| 25  | 2.02±02          | 2.00±02          | 2.00±02          | 2.00±02          | 2.00±02          |
| 26  | 1.00±02          | 1.00±02          | 1.00±02          | 1.00±02          | 1.00±02          |
| 27  | 2.53±02          | 2.05±02          | 2.00±02          | 2.00±02          | 2.00±02          |
| 28  | 3.52±02          | 2.00±02          | 2.00±02          | 2.00±02          | 2.00±02          |
| 29  | 1.19±02          | 1.16±02          | 1.15±02          | 1.18±02          | 1.19±02          |
| 30  | 3.74±02          | 3.57±02          | 3.56±02          | 3.65±02          | 3.64±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.

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

| Algorithm | DES | MLS |
|-----------|-----|-----|
| Algorithm 1 | No  | No  |
| Algorithm 2 | No  | Yes |
| Algorithm 3 | Yes | No  |
| MLS-EDA    | Yes | Yes |
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 |
|-----|-------------|-------------|-------------|---------|
| 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   | 2.09e+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   | 5.05e+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 |

The chi-square of the Friedman test is 32.3667 and the p-value is 2.6769e-09.

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

| Algorithms | Year | Parameters settings |
|------------|------|---------------------|
| MLS-EDA    | -    | NP=10-D, size(S)=max=3-D |
| EMNAP      | 2002 | NP=1000, r=0.5 as in [15]. |
| RWGEDA     | 2019 | NP=12,D, r=0.5, e=0.05, and p=0.5 as in [37]. |
| DOLTLBO    | 2019 | NP=100, r=0.3, H=10 as in [13]. |
| VCS        | 2016 | NP=50, r=0.5, e=0.3 as in [8]. |
| CPI-JADE   | 2016 | NP=100, C=0.6, r=0.5, e=0.3, and p=0.05 as in [12]. |
| L-SHADE    | 2014 | NP=20-D, NP_max=6, r_max=18, 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 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 |
|------------------|--------|--------|--------|--------|-----|--------|--------|
| 30$^D$           | 2.1333 | 6.4000 | 3.2333 | 5.3500 | 4.3667 | 3.7500 | 2.7667 |
| 50$^D$           | 2.2000 | 6.6333 | 2.9167 | 5.2167 | 3.7333 | 4.2167 | 3.0833 |
| 100$^D$          | 2.9667 | 6.4000 | 4.8833 | 4.6833 | 3.4667 | 3.2000 | 2.4000 |
| Rank            | 2.4333 | 6.4778 | 3.6778 | 5.0833 | 3.8556 | 3.7222 | 2.7500 |

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 |
|------------------|--------|--------|--------|--------|-----|--------|--------|
| 30$^D$           | 6.1641 | 5.8709 | 5.8454 | 11.6035 | 7.6021 | 5.9852 | 5.5293 |
| T                | 17.1386 | 16.3110 | 17.9718 | 32.1196 | 20.9623 | 16.4544 | 16.4263 |
| 100$^D$          | 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 100$^D$ 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 30$^D$ and 50$^D$ tests with the smallest ranking value, while L-SHADE ranks best on the 100$^D$ 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).

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

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 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).

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

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 30$^D$, 50$^D$ and 100$^D$ 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          | -    | $NP=10D$, size($S_{\text{max}}=3D$) |
| EMNA$_g$         | 2002 | $NP=1000$, $r=0.35$ as in [15] |
| NRO              | 2019 | $NP=5D$, $P_r=0.75$, $P_b=0.1$, $freq=0.05$, $a=0.01$, $\beta=1.5$ as in [10] |
| MRDE             | 2019 | $NP=gs\cdot gn$, $gn=25$, $Max_{\text{eval}}=2$, $Min_{\text{eval}}=3$, $Max_{\text{eval}}=NP/3$, $Max_{\text{eval}}=NP$ as in [32] |
| RWGEDA           | 2019 | $NP=12D$, $r=0.5$, $a=0.05$, and $\beta=0.5$ as in [37] |
| ELSHADE-SPACMA   | 2018 | $NP=\alpha\cdot D$, $NP_{\text{init}}=18$, $p_c=1.4$, $p_m=0.3$, $p_m=0.15$ as in [53] |
| ACoS-JADE        | 2018 | $NP=8D$, $Size(A)=3NP$, $\mu=\text{AS}/2$, $c=0.1$, $p_c=0.05$, $CR_a=0.5$, $F_a=0.5$, $c=0.05$, $n=0.1$ as in [54] |
| HSUES            | 2018 | $NP=[3\cdot \ln D]-80$, $M=450$, $N=360$, $cc=0.96$, $L=20$ as in [48] |

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

| CEC 2017          | Test | Algorithms        | p-value | R+  | R-  | +/-/n |
|-------------------|------|-------------------|---------|-----|-----|-------|
| 30D test          |       | MLS-EDA vs.       |         |     |     |       |
|                   |       | EMNA$_g$          | 2.56e-06| 436 | 0  | 29/0 |
|                   |       | NRO               | 1.67e-03| 40  | 260| 3/22/4 |
|                   |       | MRDE              | 1.32e-04| 326 | 25 | 22/4 |
|                   |       | RWGEDA            | 6.06e-05| 250 | 3  | 21/26 |
|                   |       | ELSHADE-SPACMA    | 5.03e-01| 160 | 116| 13/10/6 |
|                   |       | ACoS-JADE         | 3.88e-05| 294 | 6  | 23/1/5 |
|                   |       | HSUES             | 8.31e-01| 145 | 131| 10/4/5 |
| 50D test          |       | MLS-EDA vs.       |         |     |     |       |
|                   |       | EMNA$_g$          | 6.53e-06| 426 | 9  | 26/3/0 |
|                   |       | NRO               | 8.29e-01| 193.5| 212.5| 16/12/1 |
|                   |       | MRDE              | 3.46e-05| 409 | 26 | 26/3/0 |
|                   |       | RWGEDA            | 1.67e-05| 345 | 6  | 24/2/3 |
|                   |       | ELSHADE-SPACMA    | 6.07e-01| 168 | 132| 15/9/5 |
|                   |       | ACoS-JADE         | 9.33e-06| 350 | 1  | 25/1/3 |
|                   |       | HSUES             | 8.65e-02| 108 | 243| 7/19/3 |
| 100D test         |       | MLS-EDA vs.       |         |     |     |       |
|                   |       | EMNA$_g$          | 2.56e-06| 435 | 0  | 29/0 |
|                   |       | NRO               | 6.60e-04| 375 | 60 | 24/5/0 |
|                   |       | MRDE              | 4.80e-06| 429 | 6  | 27/2/0 |
|                   |       | RWGEDA            | 5.90e-06| 427 | 8  | 28/1/0 |
|                   |       | ELSHADE-SPACMA    | 3.31e-03| 317 | 61 | 18/10/1 |
|                   |       | ACoS-JADE         | 8.85e-06| 372 | 6  | 26/3/0 |
|                   |       | HSUES             | 2.75e-01| 149 | 229| 9/20/0 |

[53], ACoS-JADE [54] and HSUES [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 HSUES 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 HSUES 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 HSUES 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 HSUES.

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 HSUES 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, HSUES similarly ranks the best in this experiment. Our MLS-EDA lags behind HSUES by only a small margin, ranking second. Benefiting from its out performance on the 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 HSUES, NRO and ELSHADE-SPACMA but surpasses RWGEDA, MRDE, ACoS-JADE and EMNA$_g$.

Although HSUES 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 | EMNAG | NRO | MRDE | RWGEDA | ELSHADE-SPACMA | ACoS-JADE | HSES |
|------------|---------|-------|-----|------|--------|----------------|------------|------|
| 30D        | 3.1724  | 7.9665 | 1.8966 | 5.2069 | 4.5517 | 3.8448 | 5.7759 | 3.5862 |
| 50D        | 2.9310  | 7.5172 | 3.6465 | 5.7586 | 4.8103 | 2.9655 | 5.9310 | 2.6207 |
| 100D       | 2.4310  | 7.7586 | 3.9655 | 5.4138 | 5.7931 | 2.9828 | 5.6724 | 1.9828 |
| SR         | 2.8484  | 7.7471 | 3.1029 | 5.4557 | 5.0517 | 3.2644 | 5.7751 | 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 | EMNAG | NRO | MRDE | RWGEDA | ELSHADE-SPACMA | ACoS-JADE | HSES |
|------------|---------|-------|-----|------|--------|----------------|------------|------|
| 30D        | 3.7701  | 3.4623 | 6.1233 | 9.9732 | 3.8259 | 3.7772 | 6.5542 | 4.1022 |
| T          | 9.8944  | 9.4636 | 16.5100 | 21.1926 | 9.9933 | 10.2285 | 11.5115 | 13.1574 |
| 100D       | 63.8625 | 59.3155 | 58.3809 | 86.6729 | 67.5865 | 66.3098 | 78.7048 | 114.9982 |
| ST         | 25.8423 | 24.0805 | 27.0047 | 39.2646 | 27.1352 | 26.7720 | 33.4581 | 44.0829 |

seen from the ST results. The basic EMNAG requires the least time, while the MLS-EDA performs more efficiently than any of the other methods except EMNAG. 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 framework, but their search mechanisms are different. In RWGEDA, the random 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 8. Multiple comparisons based on the post hoc Iman Davenport test.
FIGURE 9. Convergence performance of the MLS-EDA on the CEC 2014 30D, 50D and 100D tests.

Convergence performance on CEC 2014 30D test.
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 | EMNAA | RWGEDA | DOLTLBO | VCS | CPI-JADE | L-SHADE |
|-----|---------|-------|--------|----------|-----|----------|---------|
| 01  | 0.00e+00 | 1.85e+01 | 1.84e+01 | 1.84e+01 | 4.39e+00 | 0.00e+00 | 0.00e+00 |
| 02  | 0.00e+00 | 2.00e+01 | 3.12e+01 | 3.00e+01 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| 03  | 0.00e+00 | 2.47e+01 | 6.07e+01 | 6.32e+01 | 1.62e+01 | 0.00e+00 | 0.00e+00 |
| 04  | 0.00e+00 | 2.43e+01 | 8.58e+01 | 1.27e+01 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| 05  | 2.00e+01 | 2.00e+01 | 2.07e+01 | 2.04e+01 | 3.41e+00 | 3.54e+00 | 0.00e+00 |
| 06  | 0.00e+00 | 5.49e+01 | 2.30e+01 | 8.73e+00 | 3.41e+00 | 3.54e+00 | 0.00e+00 |
| 07  | 0.00e+00 | 2.33e+01 | 1.93e+01 | 3.38e+00 | 3.87e+00 | 0.00e+00 | 0.00e+00 |
| 08  | 1.80e+01 | 3.86e+01 | 9.57e+00 | 9.46e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| 09  | 1.00e+01 | 1.00e+01 | 1.36e+01 | 8.19e+00 | 8.19e+00 | 0.00e+00 | 0.00e+00 |
| 10  | 2.09e+02 | 8.91e+02 | 3.11e+02 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| 11  | 2.38e+02 | 3.34e+02 | 1.92e+02 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| 12  | 2.07e+02 | 9.79e+02 | 4.42e+02 | 1.91e+02 | 1.07e+02 | 0.00e+00 | 0.00e+00 |
| 13  | 2.09e+02 | 5.00e+02 | 2.98e+02 | 1.96e+02 | 5.01e+02 | 0.00e+00 | 0.00e+00 |
| 14  | 2.05e+02 | 2.49e+02 | 2.84e+02 | 2.20e+02 | 2.79e+02 | 0.00e+00 | 0.00e+00 |
| 15  | 2.05e+02 | 5.69e+02 | 3.29e+02 | 3.63e+02 | 8.58e+00 | 0.00e+00 | 0.00e+00 |
| 16  | 2.05e+02 | 1.15e+03 | 9.97e+02 | 9.97e+02 | 3.74e+00 | 0.00e+00 | 0.00e+00 |
| No. | MLS-EDA | EMNAG | RWGEDA | DOLTBO | VCS | CPI-JADE | L-SHADE |
|-----|---------|--------|--------|--------|-----|----------|---------|
| 01  | 0.0e+00 | 0.0e+00 | 1.3e+09 | 1.2e+09 | 1.3e+09 | 1.3e+09 | 1.3e+09 |
| 02  | 0.0e+00 | 0.0e+00 | 6.2e+10 | 1.2e+09 | 1.2e+09 | 1.2e+09 | 1.2e+09 |
| 03  | 0.0e+00 | 0.0e+00 | 5.6e+04 | 1.2e+09 | 1.2e+09 | 1.2e+09 | 1.2e+09 |
| 04  | 5.9e+02 | 5.9e+02 | 2.1e+03 | 2.1e+03 | 2.1e+03 | 2.1e+03 | 2.1e+03 |
| 05  | 2.0e+01 | 2.0e+01 | 2.0e+01 | 2.0e+01 | 2.0e+01 | 2.0e+01 | 2.0e+01 |
| 06  | 3.7e+05 | 3.7e+05 | 2.7e+05 | 2.7e+05 | 2.7e+05 | 2.7e+05 | 2.7e+05 |
| 07  | 0.0e+00 | 0.0e+00 | 5.8e+01 | 5.8e+01 | 5.8e+01 | 5.8e+01 | 5.8e+01 |
| 08  | 3.7e+01 | 3.7e+01 | 3.7e+01 | 3.7e+01 | 3.7e+01 | 3.7e+01 | 3.7e+01 |
| 09  | 3.6e+01 | 3.6e+01 | 3.6e+01 | 3.6e+01 | 3.6e+01 | 3.6e+01 | 3.6e+01 |
| 10  | 5.0e+05 | 5.0e+05 | 5.0e+05 | 5.0e+05 | 5.0e+05 | 5.0e+05 | 5.0e+05 |
| 11  | 4.3e+03 | 4.3e+03 | 4.3e+03 | 4.3e+03 | 4.3e+03 | 4.3e+03 | 4.3e+03 |
| 12  | 2.5e+02 | 2.5e+02 | 2.5e+02 | 2.5e+02 | 2.5e+02 | 2.5e+02 | 2.5e+02 |
| 13  | 6.0e+02 | 6.0e+02 | 6.0e+02 | 6.0e+02 | 6.0e+02 | 6.0e+02 | 6.0e+02 |
| 14  | 8.1e+03 | 8.1e+03 | 8.1e+03 | 8.1e+03 | 8.1e+03 | 8.1e+03 | 8.1e+03 |
| 15  | 7.9e+02 | 7.9e+02 | 7.9e+02 | 7.9e+02 | 7.9e+02 | 7.9e+02 | 7.9e+02 |
| 16  | 2.7e+02 | 2.7e+02 | 2.7e+02 | 2.7e+02 | 2.7e+02 | 2.7e+02 | 2.7e+02 |
| 17  | 5.5e+02 | 5.5e+02 | 5.5e+02 | 5.5e+02 | 5.5e+02 | 5.5e+02 | 5.5e+02 |
| 18  | 9.3e+02 | 9.3e+02 | 9.3e+02 | 9.3e+02 | 9.3e+02 | 9.3e+02 | 9.3e+02 |
| 19  | 3.3e+02 | 3.3e+02 | 3.3e+02 | 3.3e+02 | 3.3e+02 | 3.3e+02 | 3.3e+02 |
| 20  | 8.6e+03 | 8.6e+03 | 8.6e+03 | 8.6e+03 | 8.6e+03 | 8.6e+03 | 8.6e+03 |

| TABLE 17. | Mean errors obtained from seven algorithms on the CEC 2014 10D test. |
|------------|------------------------------------------------------------------|
| No. | MLS-EDA | EMNAG | RWGEDA | DOLTBO | VCS | CPI-JADE | L-SHADE |
| 01  | 3.2e+02 | 1.7e+02 | 1.6e+02 | 2.0e+02 | 1.7e+02 | 8.1e+01 | 2.0e+02 |
| 02  | 2.7e+03 | 1.5e+03 | 1.5e+03 | 1.0e+03 | 1.5e+03 | 1.0e+03 | 1.0e+03 |
| 03  | 1.9e+05 | 1.7e+05 | 1.7e+05 | 1.2e+05 | 1.7e+05 | 1.2e+05 | 1.2e+05 |
| 04  | 1.5e+04 | 1.5e+04 | 1.5e+04 | 1.0e+04 | 1.5e+04 | 1.0e+04 | 1.0e+04 |
| 05  | 2.0e+04 | 2.0e+04 | 2.0e+04 | 1.5e+04 | 2.0e+04 | 1.5e+04 | 1.5e+04 |
| 06  | 6.6e+03 | 6.6e+03 | 6.6e+03 | 7.8e+02 | 6.6e+03 | 7.8e+02 | 7.8e+02 |
| 07  | 6.7e+03 | 6.7e+03 | 6.7e+03 | 8.9e+02 | 6.7e+03 | 8.9e+02 | 8.9e+02 |
| 08  | 7.9e+03 | 7.9e+03 | 7.9e+03 | 9.4e+02 | 7.9e+03 | 9.4e+02 | 9.4e+02 |
| 09  | 7.9e+03 | 7.9e+03 | 7.9e+03 | 9.4e+02 | 7.9e+03 | 9.4e+02 | 9.4e+02 |
| 10  | 9.4e+02 | 9.4e+02 | 9.4e+02 | 1.0e+02 | 9.4e+02 | 1.0e+02 | 1.0e+02 |
| 11  | 1.5e+04 | 1.5e+04 | 1.5e+04 | 1.8e+03 | 1.5e+04 | 1.8e+03 | 1.8e+03 |
| 12  | 1.1e+04 | 1.1e+04 | 1.1e+04 | 1.3e+03 | 1.1e+04 | 1.3e+03 | 1.3e+03 |
| 13  | 4.2e+04 | 4.2e+04 | 4.2e+04 | 4.6e+03 | 4.2e+04 | 4.6e+03 | 4.6e+03 |
| 14  | 6.7e+03 | 6.7e+03 | 6.7e+03 | 8.2e+02 | 6.7e+03 | 8.2e+02 | 8.2e+02 |
| 15  | 2.7e+04 | 2.7e+04 | 2.7e+04 | 3.2e+03 | 2.7e+04 | 3.2e+03 | 3.2e+03 |
| 16  | 2.5e+04 | 2.5e+04 | 2.5e+04 | 2.9e+03 | 2.5e+04 | 2.9e+03 | 2.9e+03 |
| 17  | 5.1e+04 | 5.1e+04 | 5.1e+04 | 5.6e+03 | 5.1e+04 | 5.6e+03 | 5.6e+03 |
| 18  | 4.8e+04 | 4.8e+04 | 4.8e+04 | 5.3e+03 | 4.8e+04 | 5.3e+03 | 5.3e+03 |
| 19  | 3.2e+04 | 3.2e+04 | 3.2e+04 | 3.7e+03 | 3.2e+04 | 3.7e+03 | 3.7e+03 |
| 20  | 7.1e+04 | 7.1e+04 | 7.1e+04 | 7.8e+03 | 7.1e+04 | 7.8e+03 | 7.8e+03 |
TABLE 18. Comparison of the computational efficiencies of the seven algorithms on the CEC 2014 test suite.

| No. | 3OD | 10DD | 90DD | 3OD | 10DD | 90DD | 3OD | 10DD | 90DD | 3OD | 10DD | 90DD |
|-----|-----|------|------|-----|------|------|-----|------|------|-----|------|------|
| 01  | 2.22 | 6.66 | 38.7 | 2.14 | 5.81 | 31.9 | 2.42 | 6.79 | 39.7 | 6.38 | 13.7 | 91.1 |
| 02  | 1.65 | 5.05 | 31.2 | 1.48 | 4.07 | 25.3 | 1.85 | 5.20 | 33.6 | 5.33 | 13.6 | 80.3 |
| 03  | 1.62 | 4.96 | 32.1 | 1.49 | 4.51 | 32.8 | 1.82 | 5.20 | 33.6 | 5.31 | 15.2 | 81.4 |
| 04  | 1.63 | 5.36 | 32.1 | 1.70 | 4.13 | 25.1 | 1.89 | 5.03 | 32.9 | 5.27 | 11.8 | 80.6 |
| 05  | 2.05 | 6.25 | 36.4 | 2.07 | 5.14 | 29.4 | 2.19 | 6.15 | 37.6 | 5.91 | 16.0 | 87.4 |
| 06  | 27.48 | 78.69 | 325.8 | 28.69 | 78.06 | 321.0 | 27.72 | 78.37 | 326.4 | 45.82 | 125.5 | 603.0 |
| 07  | 2.01 | 6.09 | 36.4 | 2.01 | 5.24 | 29.8 | 2.27 | 6.45 | 37.5 | 5.79 | 15.2 | 88.9 |
| 08  | 1.69 | 5.14 | 17.3 | 1.52 | 5.16 | 19.4 | 4.23 | 18.1 | 8.49 | 1.99 | 5.16 | 31.3 |
| 09  | 2.07 | 6.17 | 35.8 | 1.95 | 5.90 | 21.0 | 2.15 | 6.02 | 36.7 | 5.61 | 16.2 | 78.8 |
| 10  | 1.95 | 5.00 | 20.2 | 1.81 | 3.85 | 12.3 | 2.02 | 4.91 | 21.0 | 5.35 | 43.0 | 58.1 |
| 11  | 2.35 | 6.65 | 39.1 | 2.23 | 5.76 | 33.9 | 2.57 | 6.67 | 39.1 | 6.02 | 15.2 | 92.3 |
| 12  | 8.62 | 24.79 | 110.4 | 9.51 | 23.97 | 103.6 | 8.64 | 24.89 | 111.2 | 15.99 | 13.5 | 219.8 |
| 13  | 1.79 | 5.19 | 32.1 | 1.60 | 4.16 | 25.2 | 1.81 | 5.22 | 32.9 | 5.37 | 15.6 | 80.6 |
| 14  | 1.67 | 5.03 | 33.1 | 1.57 | 4.14 | 25.3 | 1.78 | 5.19 | 32.9 | 5.09 | 15.4 | 80.6 |
| 15  | 2.11 | 6.36 | 36.4 | 2.01 | 5.35 | 28.2 | 2.18 | 6.15 | 37.6 | 5.80 | 3.48 | 8.91 |
| 16  | 2.13 | 6.29 | 36.7 | 2.08 | 5.30 | 30.0 | 2.10 | 6.12 | 37.7 | 5.63 | 14.7 | 88.9 |
| 17  | 2.28 | 7.03 | 58.0 | 2.25 | 5.53 | 31.0 | 2.26 | 6.66 | 38.9 | 5.94 | 36.7 | 92.3 |
| 18  | 0.89 | 5.76 | 33.6 | 0.78 | 4.51 | 27.0 | 0.86 | 5.49 | 34.7 | 5.43 | 14.9 | 83.6 |
| 19  | 7.38 | 20.78 | 93.1 | 7.23 | 19.57 | 86.9 | 7.16 | 20.47 | 94.4 | 13.65 | 15.3 | 186.4 |
| 20  | 2.03 | 5.91 | 34.7 | 1.78 | 4.81 | 27.9 | 2.02 | 5.72 | 35.6 | 5.38 | 18.3 | 84.4 |
| 21  | 2.12 | 6.24 | 36.1 | 1.89 | 5.08 | 29.4 | 2.09 | 6.05 | 37.1 | 5.64 | 28.5 | 86.6 |
| 22  | 2.85 | 8.18 | 43.8 | 2.74 | 7.09 | 37.0 | 2.85 | 8.01 | 44.8 | 6.85 | 20.9 | 99.8 |
| 23  | 5.76 | 13.57 | 88.8 | 4.13 | 12.87 | 82.6 | 4.26 | 13.59 | 90.0 | 8.96 | 143.2 | 178.5 |
| 24  | 30.95 | 91.59 | 421.9 | 32.36 | 90.72 | 416.4 | 30.96 | 91.44 | 422.9 | 51.85 | 33.8 | 777.0 |
| 25  | 36.60 | 91.24 | 422.2 | 31.88 | 90.46 | 414.7 | 30.92 | 90.62 | 423.7 | 51.66 | 45.7 | 767.3 |
| 26  | 8.67 | 26.85 | 141.4 | 8.96 | 26.28 | 135.5 | 8.77 | 26.40 | 141.4 | 16.28 | 28.99 | 267.4 |
| 30  | 4.32 | 13.92 | 90.2 | 4.31 | 12.95 | 84.5 | 4.36 | 13.65 | 90.6 | 9.29 | 17.24 | 181.9 |

IV. CONCLUSION
As a novel extension of EDA, the framework of our proposed MLS-EDA based on DES and MLS mechanisms is described in this paper. Based on experimental evaluation, the optimal algorithm parameter values are discussed and the efficiency of the developed search mechanisms in improving the behavior of the algorithm is verified, demonstrating that our modifications play an active role in improving the algorithm performance and avoiding premature convergence.
TABLE 20. Mean errors obtained from eight algorithms on the CEC 2017 50D test.

| No. | MLS-EDA | EMNAR | NRO | MRDE | RWGJDA | ELSHADE-SPACMA | ACoS-JADE | HSES |
|-----|---------|-------|-----|------|--------|----------------|-----------|------|
| 01  | 0.000+0.000+0.000 | 2.42e+03 | 0.000+0.000+0.000 | 2.90e+03 | 0.000+0.000+0.000 | 0.000+0.000+0.000 | 0.000+0.000+0.000 | 0.000+0.000+0.000 |
| 02  | 0.000+0.000+0.000 | 8.73e+03 | 0.000+0.000+0.000 | 4.78e+03 | 0.000+0.000+0.000 | 0.000+0.000+0.000 | 0.000+0.000+0.000 | 0.000+0.000+0.000 |
| 03  | 3.86e+03 | 3.94e+03 | 3.93e+03 | 3.67e+03 | 5.38e+03 | 3.52e+03 | 3.46e+03 | 3.59e+03 |
| 04  | 6.63e+03 | 6.01e+03 | 6.32e+03 | 6.06e+03 | 3.92e+03 | 3.92e+03 | 3.92e+03 | 3.92e+03 |
| 05  | 8.27e+02 | 7.48e+02 | 7.48e+02 | 7.48e+02 | 5.90e+02 | 5.90e+02 | 5.90e+02 | 5.90e+02 |
| 06  | 4.86e+02 | 4.86e+02 | 4.86e+02 | 4.86e+02 | 4.86e+02 | 4.86e+02 | 4.86e+02 | 4.86e+02 |
| 07  | 5.29e+02 | 5.29e+02 | 5.29e+02 | 5.29e+02 | 5.29e+02 | 5.29e+02 | 5.29e+02 | 5.29e+02 |
| 08  | 4.92e+02 | 4.92e+02 | 4.92e+02 | 4.92e+02 | 4.92e+02 | 4.92e+02 | 4.92e+02 | 4.92e+02 |
| 09  | 3.96e+02 | 3.96e+02 | 3.96e+02 | 3.96e+02 | 3.96e+02 | 3.96e+02 | 3.96e+02 | 3.96e+02 |
| 10  | 8.47e+02 | 8.47e+02 | 8.47e+02 | 8.47e+02 | 8.47e+02 | 8.47e+02 | 8.47e+02 | 8.47e+02 |

TABLE 21. Mean errors obtained from eight algorithms on the CEC 2017 100D test.

| No. | MLS-EDA | EMNAR | NRO | MRDE | RWGJDA | ELSHADE-SPACMA | ACoS-JADE | HSES |
|-----|---------|-------|-----|------|--------|----------------|-----------|------|
| 01  | 3.17e-02 | 1.13e+00 | 1.13e+00 | 1.13e+00 | 1.13e+00 | 1.13e+00 | 1.13e+00 | 1.13e+00 |
| 02  | 1.51e+00 | 1.51e+00 | 1.51e+00 | 1.51e+00 | 1.51e+00 | 1.51e+00 | 1.51e+00 | 1.51e+00 |
| 03  | 5.01e-02 | 5.01e-02 | 5.01e-02 | 5.01e-02 | 5.01e-02 | 5.01e-02 | 5.01e-02 | 5.01e-02 |
| 04  | 1.12e+00 | 1.12e+00 | 1.12e+00 | 1.12e+00 | 1.12e+00 | 1.12e+00 | 1.12e+00 | 1.12e+00 |
| 05  | 4.70e-02 | 4.70e-02 | 4.70e-02 | 4.70e-02 | 4.70e-02 | 4.70e-02 | 4.70e-02 | 4.70e-02 |
| 06  | 4.38e-02 | 4.38e-02 | 4.38e-02 | 4.38e-02 | 4.38e-02 | 4.38e-02 | 4.38e-02 | 4.38e-02 |
| 07  | 4.62e-02 | 4.62e-02 | 4.62e-02 | 4.62e-02 | 4.62e-02 | 4.62e-02 | 4.62e-02 | 4.62e-02 |
| 08  | 3.96e-02 | 3.96e-02 | 3.96e-02 | 3.96e-02 | 3.96e-02 | 3.96e-02 | 3.96e-02 | 3.96e-02 |
| 09  | 5.65e-02 | 5.65e-02 | 5.65e-02 | 5.65e-02 | 5.65e-02 | 5.65e-02 | 5.65e-02 | 5.65e-02 |
| 10  | 6.28e-02 | 6.28e-02 | 6.28e-02 | 6.28e-02 | 6.28e-02 | 6.28e-02 | 6.28e-02 | 6.28e-02 |

The optimization ability of our MLS-EDA has been benchmarked using the CEC 2014 and CEC 2017 test suites. Both experiments are carried out using 30D, 50D and 100D problems. According to the statistical results, our MLS-EDA shows high efficiency and accuracy in both test suites. Moreover, our method shows competitiveness performance relative to other top-performing algorithms.
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, NRO, MRDE, RWGEDA, ELSHADE-SPACMA, ACOS-JADE AND HSES ON THE CEC 2014 TEST SUITE
See Tables 15–17.

APPENDIX F

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, 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. 1710–1719, 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. Faris, 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, Aug. 2019.
[11] R. Tanabe and A. S. Fukunaga, “Improving the search performance of SHADE using linear population size reduction,” in Proc. IEEE Evolut. Comput. (CEC), Jul. 2014, pp. 1658–1665.
Y. Wang, Z.-Z. Liu, J. Li, H.-X. Li, and G. Y. G. Yen, “Utilizing cumulative population distribution information in differential evolution,” Appl. Soft Comput., vol. 48, pp. 329–346, Nov. 2016.

Y. Xu, Z. Yang, X. Li, H. Kang, and X. Yang, “Dynamically focused optimization with Gaussian variance scaling for Gaussian estimation of distribution and its application in the multi-UAV multi-target tracking problem,” Appl. Soft Comput., vol. 78, pp. 240–260, May 2019.

P. Larrañaga and J. A. Lozano, Estimation of Distribution Algorithms, vol. 2. Boston, MA, USA: Springer, 2002.

P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Pena, “Combining evolutionary optimization by learning and simulation of Bayesian and Gaussian networks,” in Proc. 16th Conf. Uncertain. Artif. Intell., Nov. 2000, pp. 343–352.

M. Pelikan and H. Muehlenbein, “The bivariate marginal distribution algorithm,” in Advances in Soft Computing, London, U.K.: Springer, 1999, pp. 521–535.

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.

J. Grahl, P. A. N. Bosman, and F. Rothlauf, “The cluster-based variance collapse IDEA,” in Proc. 8th Annu. Conf. Genetic Evol. Comput. (GECCO), 2006, p. 397.

P. A. N. Bosman and D. Thierens, “Adaptive variance scaling in continuous estimation of distribution algorithms,” in Proc. 9th Annu. Conf. Genetic Evol. Comput. (GECCO), 2007, p. 500.

P. A. N. Bosman, J. Grahl, and D. Thierens, “Benchmarking parameter-free AMaLaGm on functions with and without noise,” Evol. Comput., vol. 21, no. 3, pp. 445–469, Sep. 2013.

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.

Z. Ren, C. He, D. Schmehl, 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.

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.

N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” Evol. Comput., vol. 9, no. 2, pp. 159–195, Jun. 2001.

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.

I. Loshchilov, “CMA-ES with restarts for solving CEC 2013 benchmark problems,” in Proc. IEEE Congr. Evol. Comput., Jun. 2013, pp. 360–367.

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.

G. A. Jastrzebski and D. V. Arnold, “Improving evolution strategies through active covariance matrix adaptation,” in Proc. IEEE Int. Conf. Evol. Comput., Jul. 2006, pp. 2814–2821.

I. Loshchilov, M. Schonauer, 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.

B. Yuan and M. Gallagher, “On the importance of diversity maintenance in estimation of distribution algorithms,” in Proc. 4th Conf. Genetic Evol. Comput. (GECCO), 2005, p. 719.

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.

T. Miqueléz, 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.

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.
**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.

**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.

**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.

* * *