Information Sciences 606 (2022) 350–367

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A novel adaptive L-SHADE algorithm and its application in UAV swarm resource configuration problem

Yintong Li, Tong Han, Huan Zhou *, Shangqin Tang, Hui Zhao

Aviation Engineering School, Air Force Engineering University, Xi'an 710038, China

ARTICLE INFO

Article history:
Received 8 June 2021
Received in revised form 11 May 2022
Accepted 15 May 2022
Available online 18 May 2022

Keywords:
Artificial intelligence
Evolutionary algorithm
L-SHADE
UAV swarm
CEC 2018 test suite
CEC 2014 test suite

ABSTRACT

In order to further improve the performance of L-SHADE, one of the most competitive variants of differential evolution (DE), a novel adaptive L-SHADE algorithm named AL-SHADE is proposed in the study. Two main parts have been modified for L-SHADE. In one part, a novel mutation strategy current-to-*Amean*/1 is added to the mutation process to improve the exploitation ability and make full use of population information. In another part, a selection strategy with adaptation scheme for mutation strategies is proposed to tune the exploitation and exploration. The performance of AL-SHADE is evaluated using CEC 2018 and CEC 2014 test suites comparing with L-SHADE, and its state-of-the-art variants, i.e., DbL-SHADE, EB-LSHADE, ELSHADE-SPACMA, jSO, and mL-SHADE. The statistical results demonstrate that AL-SHADE outperforms other competitors in terms of convergence efficiency and accuracy. Finally, AL-SHADE is applied to solve the problem of UAV swarm resource configuration, and the promising performance of AL-SHADE for solving constrained optimization problem are demonstrated by the experimental results. The source code of AL-SHADE can be downloaded from https://github.com/Yintong-Li/AL-SHADE.

1. Introduction

The optimization algorithms and their applications have became the research focus during the past decades because many problems in science, engineering, and statistics can be modeled as optimization problems by designing suitable objective functions. Compared with traditional optimization algorithms, modern nature-inspired algorithms have been found to be effective at generating optimal or near-optimal solutions for non-separable rough optimization [1]. Success-History based Adaptive Differential Evolution (SHADE) [2], one of the most competitive variants of Differential Evolution (DE) [3], in which a history-based parameter adaptation scheme was adopted to enhance the exploitation and exploration, has been claimed to be an excellent algorithm for solving optimization problems. And in order to further enhance the performance of SHADE, the L-SHADE was proposed in [4] by introducing the Linear Population Size Reduction (LPSR) mechanism in SHADE, in which the population size was reduced in accordance with a linear function. L-SHADE demonstrated a competitive performance in 2014 IEEE Congress on Evolutionary Computation (CEC 2014) and it was the winner of CEC 2014. Thus, L-SHADE is used to solve many real-world optimization problems [5–9].

Although L-SHADE has excellent performance, there are still some drawbacks in L-SHADE. Thus, a series of improvements for L-SHADE were proposed to overcome the drawbacks during the last decade. In [10], a variant of L-SHADE named DEsPA

* Corresponding author.
E-mail address: kgy_zhouh@163.com (H. Zhou).

https://doi.org/10.1016/j.ins.2022.05.058
0020-0255/

was proposed. In DEsPA, there are three main modifications. Firstly, the external archive is no longer used. Then, the greediness factor p is adapted instead of a constant value to tune the exploration and exploitation capabilities. Finally, the population size is adapted in both directions, i.e., increase and decrease in a linear manner rather than only reduction in a linear manner, to enhance the population diversity and convergence speed. In [11], an ensemble variant of L-SHADE, named SPS-L-SHADE-EIG, was proposed, and that combined the L-SHADE with an eigenvector-based (EIG) crossover and a successful-parent selecting (SPS) framework. The EIG crossover provides superior performance on numerical optimization problems with highly correlated variables. The SPS framework provides an alternative of the selection of parents to prevent the situation of stagnation. In [12], the LSHADE-EpSin was proposed to improve the L-SHADE algorithm. In LSHADE-EpSin, an ensemble sinusoidal approach was used to automatically adapt the values of the scaling factor of the DE in order to find an effective balance between the exploitation and the exploration, and a local search method based on Gaussian Walks was also adopted at later generations to increase the exploitation ability. In [13], an improved variant of the L-SHADE algorithm, called iL-SHADE, was proposed, in which the memory update mechanism and the initial historical memory values in *MCR* were modified to improve the adaptive ability and robustness of the L-SHADE. It is widely acknowledged that the mutation operation plays a vital role in the performance of DE-based algorithm, and a comprehensive analysis of mutation strategies proposed during last two decades is presented in [14] to presents recommendations and suggestions for developing efficient DE-based algorithms. In [15], a variant of the iL-SHADE algorithm, called jSO, was proposed and the main difference between jSO and iL-SHADE was that a new weighted version of mutation strategy was adopted in jSO. In [16], firstly, the author proposed Semi-Parameter adaptation of Scaling Factor and Crossover Rate for L-SHADE, named LSHADE-SPA. Then, they proposed a variant of CMA-ES by modifying the crossover mechanism to improve the exploration capability of CMA-ES. Finally, a hybrid algorithm named LSHADE-SPACMA was proposed by combining LSHADE-SPA and the modified version of CMA-ES to make full use of each advantages and characters of the two algorithms. In [17], the ELSHADE-SPACMA was proposed in which p value is dynamic rather than constant to enhance the exploration and exploitation. Additionally, another directed mutation strategy within the hybridization framework [18] was integrated to further enhance the performance of ELSHADE-SPACMA. In [19], the DbL-SHADE was proposed in which the scaling factor and crossover rate values were calculated based on the Euclidean distance between the trial and the original individual to improve the exploration ability of L-SHADE. In [20], the population-wide inertia term (PWI) was added to mutation strategies of L-SHADE, in which the L-SHADE individuals would move towards to the average direction to improve the quality of the solution in the previous generation. In [21], the mL-SHADE was proposed, in which the author made three modifications, i.e., removal of the terminal value, addition of polynomial mutation, and proposal of a memory perturbation mechanism. And the preformation of mL-SHADE was verified using CEC2019 100-Digit Challenge. In [22], there are two new mutation strategies, i.e., *DE/current-to-ord_best/1* and *DE/current-to-ord_pbest/1*, were incorporated into SHADE and L-SHADE to enhance the exploration capability and the exploitation capability, respectably, and the EB-LSHADE had been proven to have excellent performance. In [23], the mpmL-SHADE was proposed as an extended version of mL-SHADE by introducing the evolutionary process through multiple populations and dynamic control of mutation intensity and hyper-parameters. In [24], an ensemble of L-SHADE-EpSin and L-SHADE-RSP, named L-SHADE-E, was proposed. In L-SHADE-E, L-SHADE-EpSin or L-SHADE-RSP is selected randomly to run at the beginning, and then the constituent algorithm was adopted according to the progress on fitness. In [25], the long-tailed property of the Cauchy distribution was adopted in LSHADE-RSP to increase the exploration, and the strategy could generate a trial vector in mutation step with great diversity. The above-mentioned existing variants of LSHADE improve the optimization performance of LSHADE from different aspects, but they do not make full use of the information of external archive. Since L-SHADE exhibits competitive performance and has the potential to further improve performance, a novel adaptive variant of L-SHADE, named AL-SHADE, is proposed in this work.

The rest of this paper is organized as follows: The brief review of L-SHADE is described in Section 2, and the mathematical model of AL-SHADE is described in Section 3. In Section 4, the statistical results of numerical experiment based on the CEC 2018 test suite and CEC 2014 test suite are discussed. In Section 5, AL-SHADE is adopted to solve the UAV swarm resource configuration problem and the results are discussed. Finally, Section 6 concludes the work and discusses future work.

2. Related work

Recently, DE is still selected as a basis for developing novel evolution algorithms in the continuous optimization area [26–29], even though it was developed in 1995. And L-SHADE [4] is regard as one of the most successful variants of DE over the past decades because it was the winner of IEEE Competitions in Evolutionary Computation in 2014 (IEEE CEC 2014). In this section, L-SHADE is described in details.

2.1. Initialization strategy

The initial individuals (solution vectors) in population are initialized randomly as Eq. (1).

$$\boldsymbol{x}_{i,j}^{0} = L_j + rand \cdot (U_j - L_j), \qquad i = 1, 2, 3 \cdots, NP; \qquad j = 1, 2, 3 \cdots, D \tag{1}$$

where *rand* is a uniformly distributed random number between 0 and 1. $\boldsymbol{L}$ and $\boldsymbol{U}$ are the problem-specific upper and lower bounds that define the search space. NP is the value of population size. D is the dimensionality of the problem-specific.

The values of historical memory $\boldsymbol{M}_{CR}$, $\boldsymbol{M}_F$ with H entries for the crossover rate $\boldsymbol{CR}$ and scaling factor $\boldsymbol{F}$ are all initialized to 0.5 in the initialization phase as Eq. (2). It is noted that all the elements of $\boldsymbol{CR}$ and $\boldsymbol{F}$ are both scalars between 0 and 1, and H is the size of historical memories. Moreover, the external archive $\boldsymbol{A}$ of inferior solutions is initialized to empty due to there is no previous inferior solutions.

$$\begin{cases} M_{CR,k} = 0.5 \\ M_{F,k} = 0.5 \end{cases}, k = 1, 2, 3 \cdots, H \tag{2}$$

2.2. Mutation strategy

In L-SHADE, the mutation strategy *current-to-pbest*/1 is adopted. The mutated vector $\boldsymbol{v}_i$ for individual $\boldsymbol{x}_i$ is generated using four vectors (individuals in population) as Eq. (3). And the scaling factor F_i is generated by Eq. (4).

$$\boldsymbol{v}_i = \boldsymbol{x}_i + F_i(\boldsymbol{x}_{pbest} - \boldsymbol{x}_i) + F_i(\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}) \tag{3}$$

$$F_i = randc(M_{F,r3}, 0.1) \tag{4}$$

In Eq. (3), $\boldsymbol{x}_{pbest}$ is a randomly selected dominant individual from the top $NP \times p$ individuals in current population and $p \in (0,1)$ is the greediness control parameter which tune exploitation and exploration (small p behaves more greedily). $\boldsymbol{x}_{r1}$ is randomly selected from current population, and $\boldsymbol{x}_{r2}$ is randomly selected from the union of the current population and external archive. It is noted that $\boldsymbol{x}_{r1}$ and $\boldsymbol{x}_{r2}$ differ from each other as well as $\boldsymbol{x}_i$. In Eq. (4), F_i is a random value which obeys the Cauchy distribution with mean $M_{F,\,r3}$ and standard deviation value parameter 0.1, and $M_{F,\,r3}$ is a randomly selected value from memory $\boldsymbol{M}_F$. F_i will be set to 1 if $F_i > 1$, and if $F_i \leq 0$, Eq. (4) will be repeatedly applied to generate a valid value. What is more, the correction strategy will be adopted as Eq. (5) if the mutant vector element $v_{i,\,j}$ is beyond the search range boundary $[L_j, U_j]$.

$$v_{i,j} = \begin{cases} (L_j + x_{i,j})/2, & v_{i,j} < L_j \\ (U_j + x_{i,j})/2, & v_{i,j} > U_j \end{cases} \tag{5}$$

2.3. Crossover strategy

After the mutation strategy is applied to generated mutated vector $\boldsymbol{v}_i$, the trial vector $\boldsymbol{u}_i$ is generated based on the mutated vector $\boldsymbol{v}_i$ and original vector $\boldsymbol{x}_i$ using binomial crossover strategy as Eq. (6).

$$u_{i,j} = \begin{cases} v_{i,j}, & r < CR_i \quad \text{or} \quad j = j_{rand} \\ x_{i,j}, & \text{otherwise} \end{cases}. \tag{6}$$

In Eq. (6), $r \in (0,1)$ is a uniformly distributed random number, $j_{rand} \in [1, D]$ is a decision variable index which is a uniformly randomly generated and it can ensure at least one component of trial vector $\boldsymbol{u}_i$ will be taken from the mutated vector.

The crossover rate value CR_i of the individual $\boldsymbol{x}_i$ is given from a Gaussian distribution as Eq. (7). And if CR_i generated by Eq. (7) is outside of [0, 1], it will be replaced by the limit value (0 or 1) closest to the generated value.

$$CR_i = \begin{cases} 0, & M_{CR,r3} = \perp \\ \text{randn}(M_{CR,r3}, 0.1), & \text{otherwise} \end{cases} \tag{7}$$

In Eq. (7), $\perp$ denotes terminal value. $M_{CR,\,r3}$ is the mean value parameter selected from the crossover rate historical memory $\boldsymbol{M}_{\boldsymbol{CR}}$ by the same index $r3$ as in the scaling factor F case and 0.1 is the standard deviation value parameter.

2.4. Selection strategy

After all the trial vectors $\boldsymbol{u}$ of current generation g have been generated, the selection operator based on the objective function value is applied to determine the survivors for the next generation $g + 1$ as Eq. (8).

$$\boldsymbol{x}_i^{g+1} = \begin{cases} \boldsymbol{u}_i^g, & f(\boldsymbol{u}_i^g) \leqslant f(\boldsymbol{x}_i^g) \\ \boldsymbol{x}_i^g, & otherwise \end{cases} \tag{8}$$

In Eq. (8), f is the objective function for problem-specific.

2.5. External archive and historical memory update strategy

In L-SHADE, external archive $\boldsymbol{A}$ is adopted to enhance the diversity of population and avoid premature convergence. If the parent vectors $\boldsymbol{x}_i^g$ is better than the trial vectors $\boldsymbol{u}_i^g$ it will be preserved to next generation, otherwise it will be preserved to external archive $\boldsymbol{A}$. Once the size of the archive reach to the predefined size, an element randomly selected is replaced by the newly inserted element. F_i and CR_i values that succeed in generating the trial individual, that is better than the original indi-

vidual, are regarded as successful values. All the successful values of F_i and CR_i are respectively preserved to $\boldsymbol{S}_{\mathrm{CR}}$ and $\boldsymbol{S}_F$ which are used to update historical memories $\boldsymbol{M}_{CR}$, $\boldsymbol{M}_F$ at the end of each generation as Eqs. (9)-(10). A pair of elements of $\boldsymbol{M}_{CR}$ and $\boldsymbol{M}_F$ are updated in a generation, and the index k of the updated cell starts at 1 and increases by 1 after each generation, and k will be reset to 1 if it overflows the memory size H. It is noted that $\boldsymbol{M}_{CR}$, $\boldsymbol{M}_F$ will be not updated if $\boldsymbol{S}_{\mathrm{CR}} = \boldsymbol{S}_F = \varnothing$, i.e., there is no trial vector is better than the original vector in a generation. What is more, the last entry of the memories $\boldsymbol{M}_{CR}$ and $\boldsymbol{M}_F$ always keep 0.9 during the optimization process in AL-SHADE proposed in this paper. In other words, the index k of the updated cell will be reset to 1 if it overflows H-1.

$$M_{F,k} = \begin{cases} M_{F,k}, & S_F = \varnothing \\ \mathrm{mean}_{WL}(S_F), & \text{otherwise} \end{cases} \tag{9}$$

$$M_{CR,k} = \begin{cases} M_{CR,k}, & S_{CR} = \varnothing \\ \perp, & \max(S_{CR}) = 0 \\ \mathrm{mean}_{WL}(S_{CR}), & \text{otherwise} \end{cases} \tag{10}$$

In Eqs. (9)-(10), $\mathrm{mean}_{WL}(\boldsymbol{S})$ is the weighted Lehmer mean that is computing as Eq. (11) and in which $\boldsymbol{S}$ refers to either $\boldsymbol{S}_{CR}$ or $\boldsymbol{S}_F$.

$$\mathrm{mean}_{WL}(\boldsymbol{S}) = \frac{\sum_{n=1}^{|\boldsymbol{S}|} \omega_n \cdot S_n^2}{\sum_{n=1}^{|\boldsymbol{S}|} \omega_n \cdot S_n} \tag{11}$$

$$\omega_n = \frac{|f(\boldsymbol{u}_n) - f(\boldsymbol{x}_n)|}{\sum_{n=1}^{|\boldsymbol{S}|} |f(\boldsymbol{u}_n) - f(\boldsymbol{x}_n)|} \tag{12}$$

2.6. Linear population size reduction

In L-SHADE, the Linear Population Size Reduction (LPSR) is adopted to dynamically resize the population, i.e., the population size continuously reduces to match a linear function where the population size at generation 1 and G (Number of iterations at termination) is N_{init} and N_{min}, respectively. The population size of generation $g + 1$ is updated as Eq.(13).

$$NP_{g+1} = \mathrm{round}\left[\frac{FEs}{MaxFEs}(N_{\min} - N_{\mathrm{init}}) + N_{\mathrm{init}}\right] \tag{13}$$

In Eq. (13), round () returns the nearest integer number. *FEs* is the current number of objective function evaluations and *MaxFEs* is the maximum number of objective function evaluations. In the end of the generation g, the NP_g-NP_{g+1} worst-ranking individuals are discarded from the current population. In summary, the flowchart of L-SHADE is shown in Fig. 1.

3. The proposed algorithm

In this part, the novel proposed variant of L-SHADE, named AL-SHADE, is described. The improvement strategies for L-SHADE mainly consist of two parts. In one part, a novel mutation strategy *current-to-Amean*/1 is added to the mutation process. In another part, a selection strategy with adaptation schemes for mutation strategies is adopted. AL-SHADE is described in details in this section.

3.1. Mutation strategy based on weighted mean

The mutation strategy *current-to-pbest*/1 has been proved to be an effective mutation strategy to effectively accelerate the convergence rate of DE-based algorithm, and it can also avoid falling into local optimum to a certain extent compared with the mutation strategy *current-to-best*/1. However, the number of top $NP \times p$ individuals in current population shrinks as the population reduces. In the later stages of the optimization process, $\boldsymbol{x}_{\mathrm{pbest}}$ is close to $\boldsymbol{x}_{\mathrm{best}}$, or even $\boldsymbol{x}_{\mathrm{pbest}}$ is $\boldsymbol{x}_{\mathrm{best}}$ that will lead to fall into local optimum. Moreover, these individuals preserved in $\boldsymbol{A}$ are dominant individuals that are eliminated in generation g in L-SHADE. And small number of individuals in the external archive $\boldsymbol{A}$ are randomly selected as $\boldsymbol{x}_{r2}$ for mutation in Eq. (3) that does not make full use of the group information in the optimization process. To address the above-mentioned drawback, a novel mutation strategy, named *current-to-Amean*/1, is proposed as Eq.(14) in this work.

$$\boldsymbol{v}_i = \boldsymbol{x}_i + F_i(\boldsymbol{x}_{Amean} - \boldsymbol{x}_i) + F_i(\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}) \tag{14}$$

In Eq.(14), $\boldsymbol{x}_{Amean}$ is an estimation of the global optimal solution based on the promising individuals in external archive $\boldsymbol{A}$ that will avoid falling into local optimum. And $\boldsymbol{x}_{Amean}$ is calculated by using Eq.(15).

$$x_{Amean} = \sum_{i=1}^{m} w_i \cdot x_i \tag{15}$$

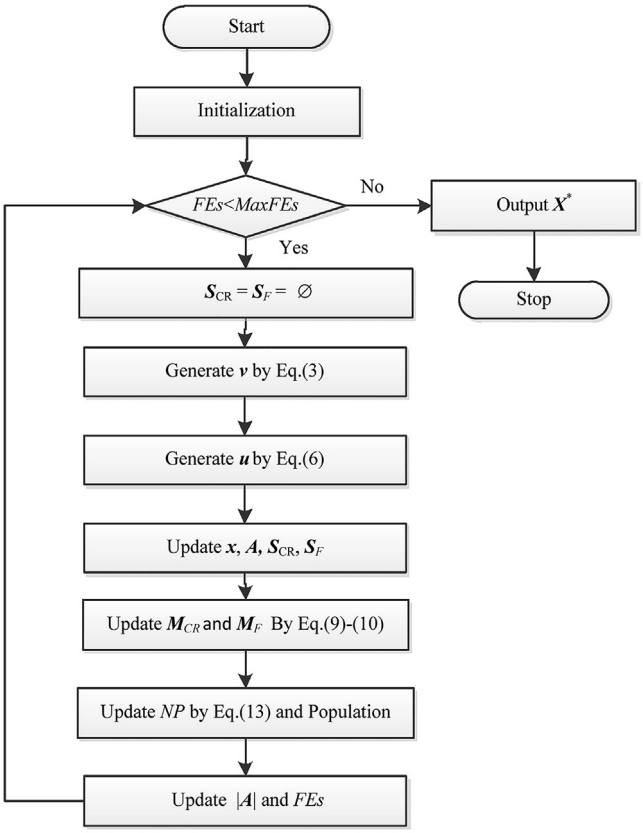

Fig. 1. The flowchart of L-SHADE.

$$w_i = \frac{\ln(m+1/2) - ln(i)}{\sum_{i=1}^{m}(\ln(m+1/2) - ln(i))} \tag{16}$$

$$m = \text{round}(e \cdot |\boldsymbol{A}|) \tag{17}$$

In Eqs. (15)-(17), the m individuals with better fitness in external archive $\boldsymbol{A}$ are selected as the promising population that are used to calculate the weighted mean $\boldsymbol{x}_{Amean}$. w_i is the weight of x_i calculated by Eq. (16), and $x_1, x_2, x_3, \ldots, x_{\text{m}}$ are the m promising individuals with objective function values ranked from high to low, i.e., x_1 is the best one and x_{m} is the worst one among $x_1, x_2, x_3, \ldots, x_{\text{m}}$. $|\boldsymbol{A}|$ is the number of individuals in external archive $\boldsymbol{A}$ and $e \in (0, 1)$ is the elite factor parameter.

In addition, the update mechanism of external archive $\boldsymbol{A}$ is adjusted in AL-SHADE to make full use of the newly generating dominant individuals rather than the previous dominant individuals, i.e., the trial vectors $\boldsymbol{u}_i^g$ that is better than the parent vectors $\boldsymbol{x}_i^g$ will be preserved to next generation and external archive $\boldsymbol{A}$. Since the individuals in $\boldsymbol{A}$ are used in the first generation, $\boldsymbol{A}$ is no longer initialized to an empty set but the best individual generated by the initialization is saved in $\boldsymbol{A}$.

3.2. Selection strategy with adaptation schemes

It is noted that the novel mutation strategy *current-to-Amean*/1 is added as one of the mutation strategies rather than directly replace the mutation strategy *current-to-pbest*/1 because Eq. (3) and Eq. (14) play different roles at different stages of optimization. It is vital to effectively integrate the two mutation strategies to fully exploit the advantages of them. Therefore, the selection strategy with adaptation schemes is proposed in this section to effectively integrate the two mutation strategies. There is an adaptive probability parameter P_S that is adopted to select Eq. (3) or Eq. (14) with a certain probability for each individual in each generation. Because there is no prior knowledge, P_S is initialized to 0.5 and adaptively adjusted in search process as Eq. (18).

$$P_S^{g+1} = P_S^g + \frac{0.05 \cdot (1 - P_S^g) \cdot (P_1 - P_2) \cdot FEs}{MaxFEs} \tag{18}$$

$$P_1 = \frac{M1_{better}}{M1} \tag{19}$$

$$P_2 = \frac{M2_{better}}{M2} \tag{20}$$

In Eqs. (18)-(20), $M1$ and $M2$ are the number of individuals using Eq. (3) and Eq. (14) as the mutation strategy in g^{th} generation, respectively. $M1_{better}$ and $M2_{better}$ are the number of better individuals generated by using Eq. (3) and Eq. (14) as the mutation strategy in g^{th} generation. Thus, P_1 and P_2 are the probability of using Eq. (3) and Eq. (14) as the mutation strategy to produce a better solution in g^{th} generation, respectively. Moreover, if Ps^{g+1} is outside of [0.1, 0.9], it will be replaced by the limit value (0.1 or 0.9) closest to the generated value.

3.3. The framework of AL-SHADE

In summary, the flowchart of the proposed algorithm is shown in Fig. 2. The main difference between AL-SHADE and LSHADE includes the novel mutation strategy, the selection strategy with adaptation schemes, and the update mechanism of external archive $\boldsymbol{A}$. For each solution, if the random number rand < P_S, Eq. (3) will be selected to generate the mutated vector $\boldsymbol{v}_i$, and conversely Eq. (14) will be selected. The MATLAB source code of AL-SHADE can be downloaded from https://github.com/Yintong-Li/AL-SHADE.

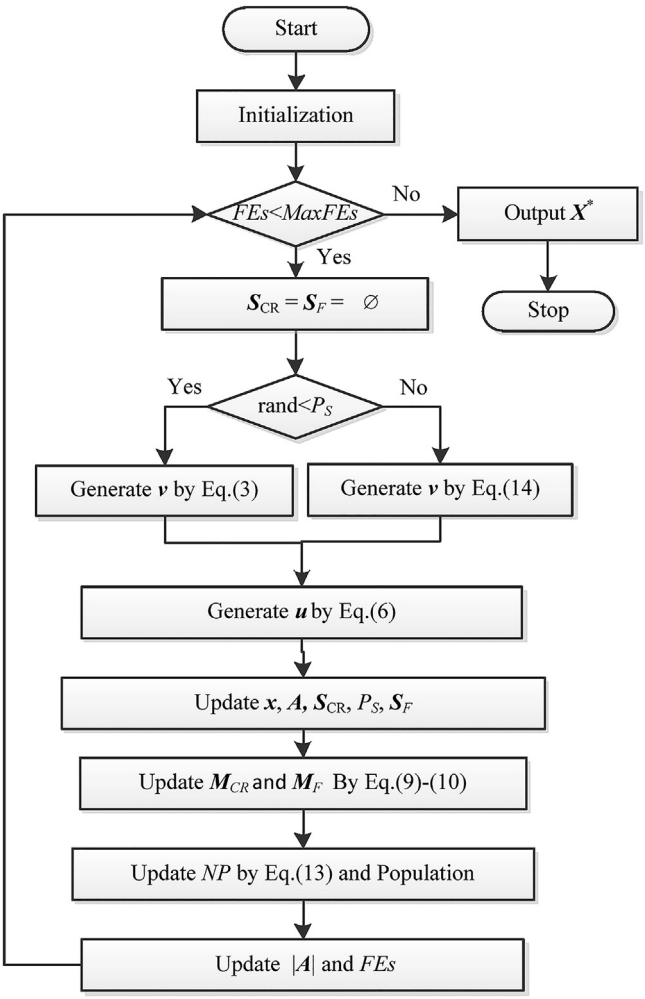

Fig. 2. The flowchart of the proposed algorithm.

4. Numerical experiment based on CEC2018 and CEC2014 test suites

In this section, in order to investigate the performance of AL-SHADE, the comprehensive experiments based on CEC 2018 test suite [30] and CEC 2014 test suite [31] on single objective bound constrained real-parameter optimization have been conducted. It is noted that F2 included in CEC 2017 test suit has been excluded because it shows unstable behavior and the remaining 29 benchmark functions in CEC 2017 test suite are used as CEC 2018 test suite. Details of the CEC 2018 test suite and CEC 2014 test suite can refer to [30,31].

4.1. Experiment environment and parameter settings

As noted in [30,31], the search range is $[-100,100]^D$ and the maximum function evaluations (*MaxFEs*) is set to $D \times 10{,}000$ for all benchmark functions. D is the dimensionality of the test problem. In this work, *MaxFEs* is set to 100,000, 300,000, and 500,000 for D = 10, 30, and 50 in the experiments based on CEC 2018 test suite and CEC 2014 test suite. Each benchmark function is independently solved 51 times to reduce the randomness. The error measure, defined as $f(\boldsymbol{x}_{Best})$-$f(\boldsymbol{x}^*)$, obtained by each algorithm is recorded as the result for convenience, where $\boldsymbol{x}_{Best}$ is the solution obtained by the algorithm and $\boldsymbol{x}^*$ is the global optimum solution of the benchmark function. It is noted that the error measure value smaller than 10^{-8} will be taken as 0.

The performance of AL-SHADE is evaluated in comparison to the state-of-the-art algorithms, including L-SHADE [4], jSO [15], ELSHADE-SPACMA [17], which are the winners of CEC 2014, 2017, and 2018, respectively. Moreover, three recently proposed variants, named DbL-SHADE [19], mL-SHADE [21], and EBLSHADE [22] are also used as comparison algorithms in the paper. It is noted that the six selected competitors except L-SHADE are all variants of L-SHADE. To make a fair comparison, the parameters of competitors are set as the original work and tabulated in Table 1. All the experiments are implemented in MATLAB R2019b and the test environment is set up on a computer with Inter(R) Core (TM) i7-10750H CPU@2.60 GHz 16 GB RAM.

In this paper, there are two non-parametric statistical hypothesis tests that are used to analyze the results derived from AL-SHADE and six competitors from a statistical point of view. Firstly, the Wilcoxon signed-rank test is adopted to check the differences between AL-SHADE and all competitors for each function. According the results of the Wilcoxon signed-rank test, it can be known whether there is significant difference between the performance of AL-SHADE and each competitor on each function, and further judge which algorithm has better performance. Then, the Friedman test is used to test differences among all seven algorithms for all functions. According the results of the Friedman test, it can be seen whether there is significant difference among the performance of all algorithms, and the final rankings of different algorithms for all functions is obtained.

4.2. Strategies effectiveness analysis

It is very necessary to evaluate the effect of the proposed strategies on the performance of AL-SHADE algorithm. The strategies effectiveness analysis will be carried out in this section. The AL-SHADE version without the proposed adaptive selection scheme, named AL-SHADE-1, will be experimentally investigated. In addition, the adaptive selection scheme was proposed to select Eq. (3) or Eq. (14) to update individual. In other words, if the proposed novel mutation strategy were not included in AL-SHADE, the adaptive selection scheme would also be not included in AL-SHADE. Therefore, there is no the AL-SHADE version without the proposed mutation strategy. It is noted that the original mutation strategy Eq. (3) and the proposed mutation strategy Eq. (14) are randomly adopted to generate an individual in AL-SHADE-1. Therefore, the effect of the proposed novel mutation strategy on the performance of LSHADE can be analyzed by comparing AL-SHADE-1 and L-SHADE, and the effect of the proposed adaptive selection scheme on the performance of AL-SHADE can be analyzed by comparing AL-SHADE and AL-SHADE-1. And, the Friedman test is used to compare all of them in this section.

The detailed statistical results, including mean value (Mean) and standard deviation (SD) of the obtain results, derived by AL-SHADE-1 on CEC2018 and CEC2014 test suites (D = 10, 30, 50) are illustrated in Table S1 in the supplementary material file. And the detailed statistical results derived by AL-SHADE and L-SHADE on CEC2018 and CEC2014 test suite (D = 10, 30, 50) are illustrated in Tables S2-S7 in the supplementary material file. In this section, the statistical analysis results of the

Table 1
Parameter settings for the seven algorithms.

Algorithms	Parameters setting
AL-SHADE	NP_{init} = 18·D, NP_{min} = 4, $\|\boldsymbol{A}\|$ = 2.6·NP, p = 0.11, H = 6, e=0.5
DbL-SHADE	NP_{init} = 100, NP_{min} = 20, H = 10, $\|\boldsymbol{A}\|$ = NP, p_{min} = 2/NP as in [19]
EB-LSHADE	NP_{init} = 18·D, NP_{min} = 4, $\|\boldsymbol{A}\|$ = 1.4·NP, p = 0.11, H = 5, c = 0.8 as in [22]
ELSHADE-SPACMA	NP_{init} = 18·D, NP_{min} = 4, H = 5, $\|\boldsymbol{A}\|$ = 1.4·NP, F_{CP} = 0.5, c = 0.8, *threshold* = *MaxFEs*/2, p_{AGDE} = 0.1, p_{init} = 0.3, p_{min} = 0.15 as in [17]
jSO	NP_{init} = 25log(D)$\sqrt{D}$, NP_{min} = 4, $\|\boldsymbol{A}\|$ = 2.6·NP, H = 5, p_{max} = 0.25, p_{min} = p_{max}/2 as in [15]
L-SHADE	NP_{init} = 18·D, NP_{min} = 4, $\|\boldsymbol{A}\|$ = 2.6·NP, p = 0.11, H = 6 as in [4]
mL-SHADE	NP_{init} = 18·D, NP_{min} = 4, $\|\boldsymbol{A}\|$ = NP, p = 0.11, H = 6, m_r = 0.05, p_m = 1/D, η = 10, N^{stuck} = 400 as in [21]

Table 2
The rankings of three algorithms in CEC 2018 and CEC 2014 test suites according to the Friedman test (α = 0.05).

Algorithms	CEC 2018 test suite			CEC 2014 test suite			Mean Ranking	Rank
	10D	30D	50D	10D	30D	50D		
AL-SHADE	1.91	1.81	**1.81**	**1.72**	**1.71**	2.00	**1.83**	**1.00**
AL-SHADE-1	**1.74**	**1.64**	1.97	1.86	1.86	**1.97**	1.84	2.00
L-SHADE	2.34	2.55	2.22	2.41	2.43	2.03	2.33	3.00
Friedman-p-value	2.81E-02	3.32E-04	2.53E-01	1.03E-02	7.16E-03	9.62E-01	N/A	N/A

Friedman test among AL-SHADE, AL-SHADE-1, and L-SHADE will be discussed to evaluate the effect of the proposed strategies on the performance. The rankings of AL-SHADE, AL-SHADE-1, and L-SHADE in CEC 2018 and CEC 2014 test suites according to the Friedman test with significance level α = 0.05 are listed in Table 2.

From Table 2, the *Friedman-p-value* for all dimensions except 50*D* in CEC 2018 and CEC 2014 test suites are less than α, i.e., there is a significant difference among the three algorithms in terms of the performance. Specifically, AL-SHADE obtains the best ranking followed by AL-SHADE-1 and L-SHADE in 50*D* in CEC 2018 test suite, 10*D*, and 30*D* in CEC 2014 test suite. This observation confirms the positive impact of the selection strategy with adaptation schemes and the novel mutation strategy in developing AL-SHADE. As for 10*D* and 30*D* in CEC 2018 test suite, and 50*D* in CEC 2014 test suite, AL-SHADE-1 gets the first ranking followed by AL-SHADE-1 and L-SHADE. This observation confirms that there are positive impact of the novel mutation strategy and negative impact of the selection strategy with adaptation schemes in developing AL-SHADE. According to the "Mean Ranking", AL-SHADE is superior to AL-SHADE-1 and L-SHADE, and AL-SHADE-1 is superior to L-SHADE. It can be concluded that the selection strategy with adaptation schemes and the novel mutation strategy have positive impact in developing AL-SHADE. Moreover, it should also be objectively seen that the selection strategy with adaptation schemes may have an opposite effect on different dimensions, which provides the guidance that AL-SHADE-1 may be better for solving simple problems, and AL-SHADE may be better for solving complex problems.

4.3. Comparison with state-of-the-art algorithms

The detailed statistical results, including mean value (Mean) and standard deviation (SD) of the obtained results, derived by AL-SHADE and the six above-mentioned competitors on CEC 2018 and CEC 2014 test suites (D = 10, 30, 50) are illustrated in Tables S2-S7 in the supplementary material file. The best solutions according to the Mean among the seven algorithms are showed at **bold**.

4.3.1. Analysis of the Wilcoxon signed-rank test results

The Wilcoxon signed-rank test is adopted to check the differences between AL-SHADE and all competitors for each function. The detailed results of Wilcoxon signed-rank test with a significance level α = 0.05 between AL-SHADE and competitors using CEC 2018 and CEC 2014 test suites for all dimensions are listed in Tables S2-S7 in the supplementary material file.

The statistical analysis results of the Wilcoxon signed-rank test with significance level α = 0.05 test between AL-SHADE and competitors using CEC2018 and CEC2014 test suites for all dimensions are summarized in Table 3. In Table 3, the number of '+' represents the number of functions in which AL-SHADE is superior to the competitor, whereas '−' indicates poorer performance. And '=' indicates that AL-SHADE is statistically like the competitor.

According to the results using CEC 2018 test suite summarized in Table 3, AL-SHADE has higher counts of '+' than '−' in comparison with six competitors on 10*D*, which implies that AL-SHADE performs significantly better than all competitors. For 30*D*, AL-SHADE performs significantly better than all competitors except EB-LSHADE, and the performance of AL-SHADE is statistically like that of EB-LSHADE. For 50*D*, AL-SHADE is superior to DbL-SHADE, L-SHADE, and mL-SHADE, is inferior to EB-LSHADE, ELSHADE-SPACMA, and jSO. According to the value of "Total" that represents the overall performance of algorithms on CEC 2018 test suite, AL-SHADE has higher total of '+' than '−' in comparison with six competitors. Thus, it can be concluded that AL-SHADE shows better performance than DbL-SHADE, EB-LSHADE, ELSHADE-SPACMA, jSO, L-SHADE, and mL-SHADE on the CEC 2018 test suite.

Table 3
The results of Wilcoxon signed-rank test between AL-SHADE and competitors (α = 0.05).

R (+/−/=) AL-SHADE v.s.	CEC 2018 test suite				CEC 2014 test suite			
	10*D*	30*D*	50*D*	Total	10*D*	30*D*	50*D*	Total
DbL-SHADE	10/2/17	24/0/5	24/1/4	58/3/26	17/2/11	18/3/9	19/6/5	54/11/25
EB-LSHADE	8/1/20	9/9/11	4/8/17	21/18/48	9/2/19	7/5/18	5/10/15	21/17/52
ELSHADE-SPACMA	8/2/19	10/7/12	10/13/6	28/22/37	10/4/16	10/10/10	6/18/6	26/32/32
jSO	14/0/15	12/5/12	8/12/9	34/17/36	13/2/15	12/9/9	9/13/8	34/24/32
L-SHADE	15/1/13	13/1/15	11/3/15	39/5/43	8/3/19	15/2/13	5/5/20	28/10/52
mL-SHADE	16/2/11	19/3/7	23/3/3	58/8/21	17/2/11	17/3/10	16/4/10	50/9/31

According to the results using CEC 2014 test suite summarized in Table 3, AL-SHADE has higher counts of '+' than '−' in comparison with six competitors on 10*D*, which implies that AL-SHADE performs significantly better than all competitors. For 30*D*, AL-SHADE performs significantly better than DbL-SHADE, EB-LSHADE, jSO, L-SHADE, and mL-SHADE, and the performance of AL-SHADE is statistically like that of ELSHADE-SPACMA. For 50*D*, AL-SHADE shows similar performance to L-SHADE, and it is superior to DbL-SHADE and mL-SHADE, but it is inferior to EB-LSHADE, ELSHADE-SPACMA, and jSO. According to the value of "Total" on CEC 2014 test suite, AL-SHADE has higher total of '+' than '−' in comparison with six competitors except ELSHADE-SPACMA. Thus, it can be concluded that AL-SHADE shows better performance than DbL-SHADE, EB-LSHADE, jSO, L-SHADE, and mL-SHADE, but is inferior to ELSHADE-SPACMA on the CEC 2018 test suite.

4.3.2. Analysis of the Friedman test results

The Friedman test is adopted to further illustrate differences among all seven algorithms for all functions. The results of Friedman test with significance level $\alpha = 0.05$ among all seven algorithms using CEC2018 and CEC 2014 test suites for all dimensions are listed in Table 4. In Table 4, "Mean" represents the mean of Friedman test results of the algorithm on 10*D*, 30*D*, and 50D, and "Rank" represents the sorting results of "Mean" values. From Table 4, the *Friedman-p-value* for all dimensions except 10*D* in CEC 2018 test suite are less than α, i.e., there is a significant difference among seven algorithms in terms of the performances in CEC 2018 and CEC 2014 test suites on all dimensions except 10*D* in CEC 2018 test suite. Additionally, the ranks for algorithms according to the Friedman test results listed in Table 4 are presented in Fig. 3 to show the results more visually.

According to the results using CEC 2018 test suite shown in Fig. 3, AL-SHADE performs better than all competitors on 10*D*. For 30*D*, AL-SHADE performs significantly better than all competitors except EB-LSHADE, and AL-SHADE is slightly inferior to EB-LSHADE. For 50*D*, AL-SHADE is superior to DbL-SHADE, L-SHADE, and mL-SHADE, is inferior to EB-LSHADE, ELSHADE-SPACMA, and jSO. According to the values of "Mean" that represents the overall performance of algorithms on CEC 2018 test suite, AL-SHADE gets the first ranking. Thus, it can be concluded that AL-SHADE outperforms DbL-SHADE, EB-LSHADE, ELSHADE-SPACMA, jSO, L-SHADE, and mL-SHADE on the CEC 2018 test suite that is consistent with the analysis results of the Wilcoxon signed-rank test in Section 4.3.1.

According to the results using CEC 2014 test suite shown in Fig. 3, AL-SHADE outperforms all competitors on 10*D* and 30*D*. For 50*D*, AL-SHADE is superior to DbL-SHADE and mL-SHADE, but is inferior to EB-LSHADE, ELSHADE-SPACMA, jSO, and LSHADE. According to the values of "Mean" on CEC 2014 test suite, AL-SHADE also gets the first ranking. Thus, it can be concluded that AL-SHADE outperforms all competitors on the CEC 2018 test suite that is consistent with the analysis results of the Wilcoxon signed-rank test in Section 4.3.1.

Table 4
Average ranks for seven algorithms according to the Friedman test ($\alpha = 0.05$).

Algorithms	CEC 2018 test suite					CEC 2014 test suite				
	10*D*	30*D*	50*D*	Mean	Rank	10*D*	30*D*	50*D*	Mean	Rank
AL-SHADE	3.21	3.02	3.33	3.18	1	2.55	3.12	3.97	3.21	1
DbL-SHADE	4.34	5.86	6.29	5.50	7	4.92	5.22	5.33	5.16	7
EB-LSHADE	3.79	2.98	2.98	3.25	2	3.30	3.65	3.33	3.43	3
ELSHADE-SPACMA	4.14	3.60	3.05	3.60	4	3.92	3.28	2.75	3.32	2
jSO	3.93	3.45	3.02	3.47	3	4.78	3.60	3.70	4.03	5
L-SHADE	4.09	3.90	3.50	3.83	5	3.70	3.98	3.68	3.79	4
mL-SHADE	4.50	5.19	5.83	5.17	6	4.83	5.15	5.23	5.07	6
Friedman-p-value	2.26E-01	3.03E-09	4.56E-15	N/A	N/A	1.26E-06	1.02E-05	1.13E-06	N/A	N/A

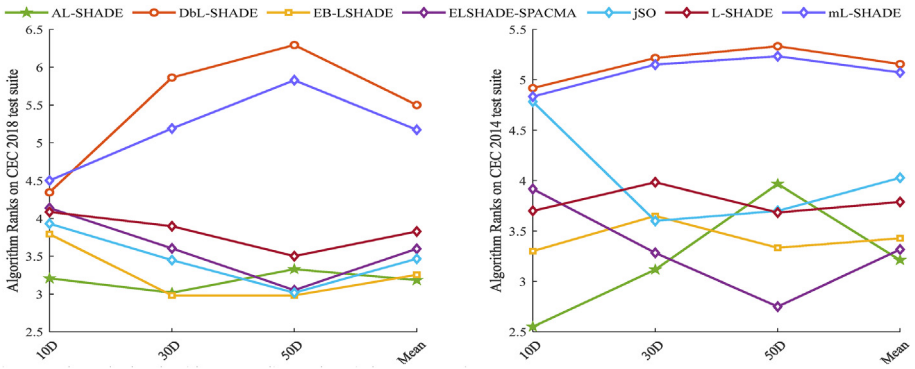

Fig. 3. The ranks for algorithms according to the Friedman test results.

What is more, the post hoc Iman-Davenport test [32] is adopted to further analyze the magnitude of significant differences. Iman-Davenport test is a statistics distributed according to the F-distribution with (K-1) and (K-1)·(N-1) degrees of freedom (DoF). K is the algorithms number and N is the number of benchmark functions, i.e., K = 7 and N = 29 for CEC2018 test suite, K = 7 and N = 30 for CEC2014 test suite. The Nemenyi test is used as the post hoc test in which the critical difference value (CDV) is used to determine the difference among seven algorithms based on the ranks obtained by the Friedman test. The CDV is calculated by using Eq. (22).

$$F_F^2 = \frac{(N-1)\cdot\chi_F^2}{N\cdot(K-1)-\chi_F^2} \tag{21}$$

$$CDV = q_a \cdot \sqrt{\frac{K\cdot(K+1)}{6N}} \tag{22}$$

where the critical value q_a is obtained from the statistical table of the F-distribution. For the CEC2018 test suite, q_a = 2.45, the CDV is 1.39 calculated by using Eq. (22). For the CEC2014 test suite, q_a = 2.43, the CDV is 1.36 calculated by using Eq. (22).

The magnitude of significant differences among seven algorithms are shown in Fig. 4. The algorithms with no significant differences can be connected using the CDV. As shown in Fig. 4, there is no significant difference between the seven algorithms on 10D in CEC2018 test suite. As for 30D and 50D in CEC2018 and CEC2014 test suites, there is no significant difference among AL-SHADE, EB-LSHADE, ELSHADE-SPACMA, jSO, and LSHADE, but AL-SHADE is significantly better than DbL-SHADE and mL-SHADE. As for 10D in CEC2014 test suite, there is no significant difference among AL-SHADE, EB-LSHADE, and LSHADE, but AL-SHADE is significantly better than ELSHADE-SPACMA, jSO, DbL-SHADE and mL-SHADE.

In conclusions, the proposed AL-SHADE has superior performance to six competitors on the CEC 2018 and CEC 2014 test suites according to the statistical analysis using the Wilcoxon signed-rank test and Friedman test.

4.3.3. Analysis of the convergence speed

To further illustrate the convergence speed of AL-SHADE, the mean convergence graphs based on 51 independent runs of seven algorithms on the sample functions (unimodal function F1, simple multimodal functions F6 and F9, hybrid functions F13 and F16, and composition function F21 in CEC 2018 test suite on 30D) are shown as Fig. 5, and the rest figures for CEC 2018 test suite on all dimensions shown as Figs. S1-S6 that could be found in the supplementary material file. It is noted that the value of mean convergence graph smaller than 10^{-8} will be taken as 0. From Fig. 5, there is no significant difference in convergence speed on the six functions between AL-SHADE and all competitors. Specifically, the convergence speed of AL-SHADE is slightly faster than that of ELSHADE-SPACMA, jSO, and mL-SHADE, but slightly slower than that of EB-LSHADE and DbL-SHADE. Thus, it can be concluded that AL-SHADE provides a promising performance in terms of convergence speed for CEC 2018 test suite.

The mean convergence graphs based on 51 independent runs of seven algorithms on the sample functions (unimodal function F1, simple multimodal functions F6 and F13, hybrid functions F17 and F22, and composition function F30 in CEC 2014 test suite on 30D) are shown as Fig. 6, and the rest figures for CEC 2014 test suite on all dimensions shown as

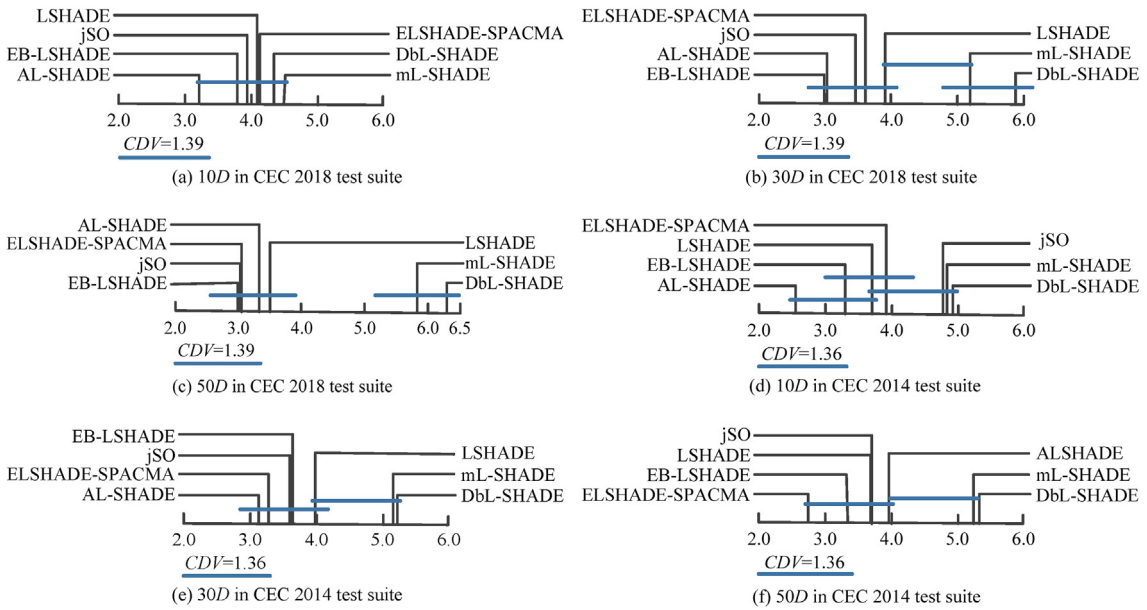

Fig. 4. Multiple comparison using CDV to connect algorithms with no significant differences.

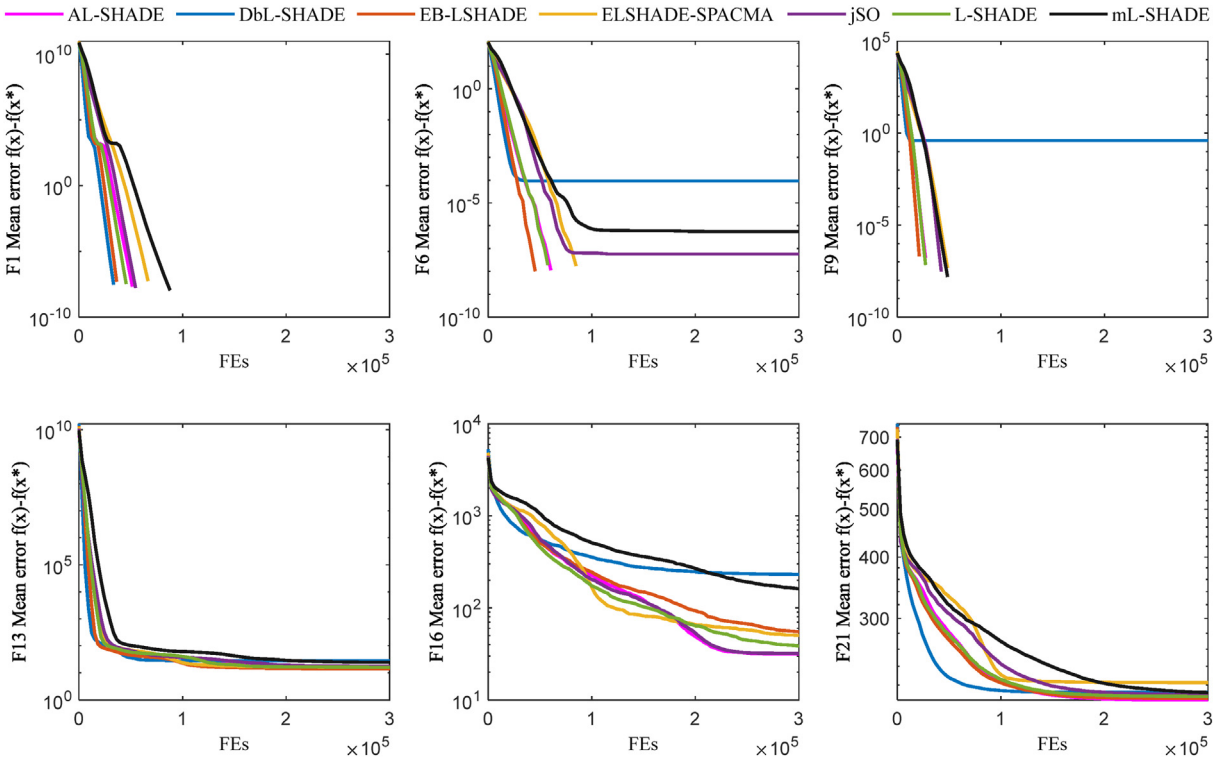

Fig. 5. Convergence graphs of seven algorithms on F1, F6, F9, F13, F16, and F21 in CEC 2018 test suite (30*D*).

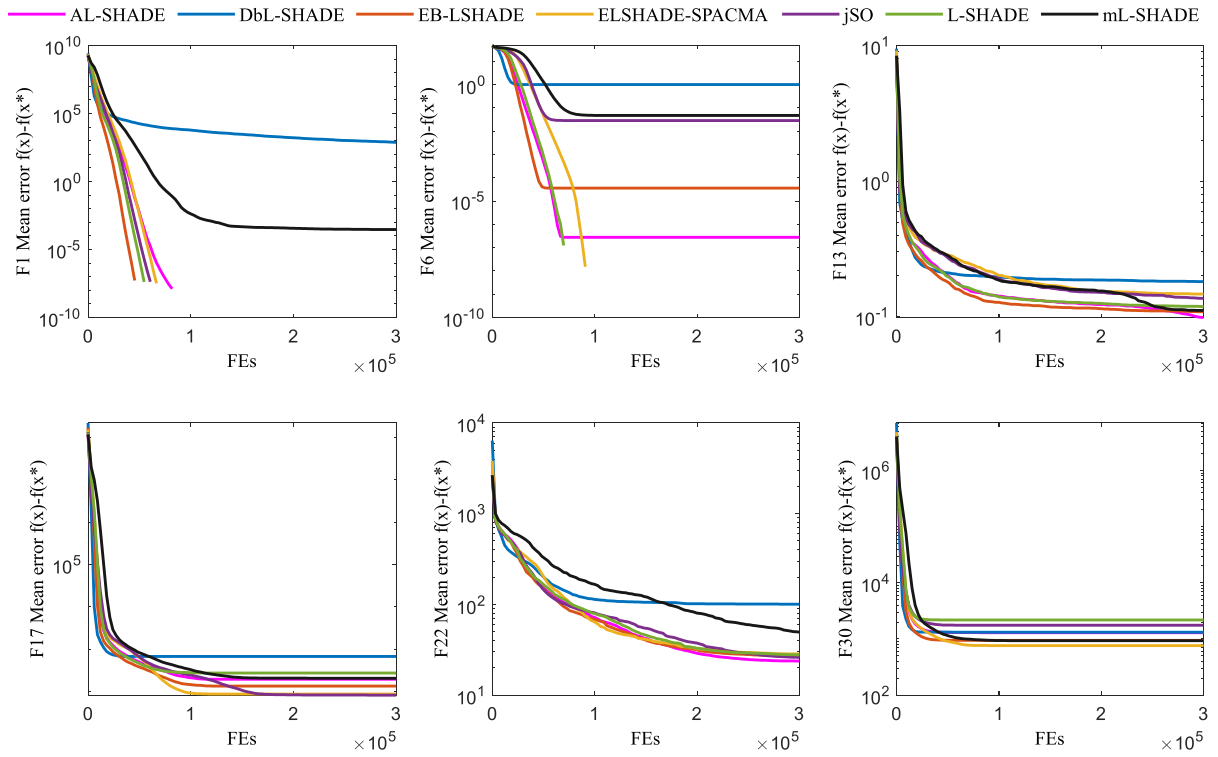

Fig. 6. Convergence graphs of seven algorithms on F1, F6, F13, F17, F22, and F30 in CEC 2014 test suite (30*D*).

Figs. S7-S12 that could be found in the supplementary file. It is noted that the value of mean convergence graph smaller than 10^{-8} will be taken as 0. From Fig. 6, the convergence speed of AL-SHADE is faster than that of ELSHADE-SPACMA, jSO, and mL-SHADE on simple multimodal functions F6. And there is no significant difference in convergence speed on the rest five functions between AL-SHADE and all competitors. Thus, it can be concluded that AL-SHADE also provides a promising performance in terms of convergence speed for CEC 2014 test suite.

4.4. Algorithm complexity

The algorithm complexity is another important term for evaluating algorithms. In this section, the instruction for evaluating algorithm complexity in the CEC2018 is adopted. There are three variables, i.e., T_0, T_1, and T_2 are employed in the evaluation, where T_0 is the time consumption by running the code shown in Table 5, T_1 denotes the computing time of 200,000 function evaluations for F18 of D dimensions from CEC 2018 test suit, and T_2 is complete computing time for the algorithm with 200,000 evaluations of the same D dimensional F18 from CEC 2018 test suit, and $\bar{T}_2$ is the mean T_2 value of 5 runs. Then, the complexity of the algorithm is reflected by $(\bar{T}_2 - T_1)/T_0$. It is noted that all parameters except NP_{init} of all algorithms are set as shown in Table 1, and NP_{init} is set to 500 for all algorithms on all dimensions in this section for the fairness and rationality of computational complexity comparison.

The algorithm complexities of seven algorithms on 10D, 30D, and 50D are listed in Table 6 and presented in Fig. 7. The proposed AL-SHADE is more time consuming than DbL-SHADE and L-SHADE on all dimensions, that is worthwhile because of the significant improvement in optimization performance. It can also be observed that the algorithm complexity of AL-SHADE is linearly related to the dimension, and the proposed strategies in this paper does not increase the algorithm complexity. That is to say that AL-SHADE performs more efficiently than the four variants of L-SHADE. It can be analyzed that the main reason why the average time consumed by AL-SHADE is more than L-SHADE is that the calculation process of the weighted mean is added in the mutation stage.

5. Solving the UAV swarm resource configuration problem by using AL-SHADE

With the development of miniaturization technology, new sensors, embedded control systems, communications technology, and artificial intelligence, the application of unmanned aircraft vehicle (UAV) swarms becomes a reality in many civilian and military missions [33]. The intelligent UAV swarm operations, as a new combat concept that changes the rules of the future battlefield, is considered a typical combat style of future intelligent warfare. The key issues of UAV swarm operations have gradually been getting more and more attention from researchers in this field. There are some literatures on the issue of UAV swarm operations involving UAV swarm coordination [34,35], cooperative mission planning for searching and attacking the time-sensitive moving targets [36], patrolling task planning under the uncertain and dynamic environment [37], dynamic UAV swarm versus UAV swarm combat [38], cooperative persistent surveillance [39]. Cooperative target search in adversarial environment [40], task and resource dynamic assignment algorithm [41,42], and so on.

The intelligent UAV swarm operation refers to deploy the swarm of intelligent UAV to carry out combat operations, in which the functions unit in OODA combat theory [43] are distributed to a certain number of small, low-cost, heterogeneous intelligent UAV platforms. And these heterogeneous intelligent UAV platforms are quickly and dynamically combined into several self-adaptive killing chains to carry out specific combat tasks through the network information system, as shown in the Fig. 8. The heterogeneous UAV swarm based on the OODA combat theory typically consists of four types of UAV, namely observation node, orientation node, decision node, and action node. The observation node is the UAV with battlefield Intelligence, Surveillance, and Reconnaissance (ISR) capabilities. The orientation node is the UAV that could judge and process the information sensed by the observation node, and conduct situational awareness. The decision node is the UAV that can make decisions based on the battlefield situation. And the action node is the UAV with firepower strikes, electromagnetic interference, and other functions, which directly executes action made by the decision node.

Table 5
The code for calculating the time T_0.

```
1. Input: t = 1,000,000
2. tic
3. x = 0.55;
4. for i = 1:t
5. x = x + x; x = x/2; x = x*x; x = sqrt(x);
6. x = log(x); x = exp(x); x = x/(x + 2);
7. end
8. T0 = toc;
9. Output: T0
```

Table 6
Computational complexity comparison of seven algorithms.

Algorithms.	T_0/s	T_1/s			$\bar{T}_2$/s			$(\bar{T}_2 - T_1)/T_0$		
		10*D*	30*D*	50*D*	10*D*	30*D*	50*D*	10*D*	30*D*	50*D*
AL-SHADE	0.01070	0.07298	0.28929	0.61960	0.32949	0.69971	1.20510	23.98326	38.37478	54.74512
DbL-SHADE	0.01070	0.07298	0.28929	0.61960	0.28634	0.59149	1.03654	19.94952	28.25598	38.98450
EB-LSHADE	0.01070	0.07298	0.28929	0.61960	0.88823	1.25433	1.80810	76.22601	90.23242	111.12588
ELSHADE-SPACMA	0.01070	0.07298	0.28929	0.61960	0.66203	1.08634	1.78693	55.07628	74.52514	109.14714
jSO	0.01070	0.07298	0.28929	0.61960	0.61642	0.92187	1.42058	50.81212	59.14642	74.89273
L-SHADE	0.01070	0.07298	0.28929	0.61960	0.30750	0.59097	1.05107	21.92784	28.20700	40.34352
mL-SHADE	0.01070	0.07298	0.28929	0.61960	0.54045	0.84955	1.25695	43.70869	52.38502	59.59303

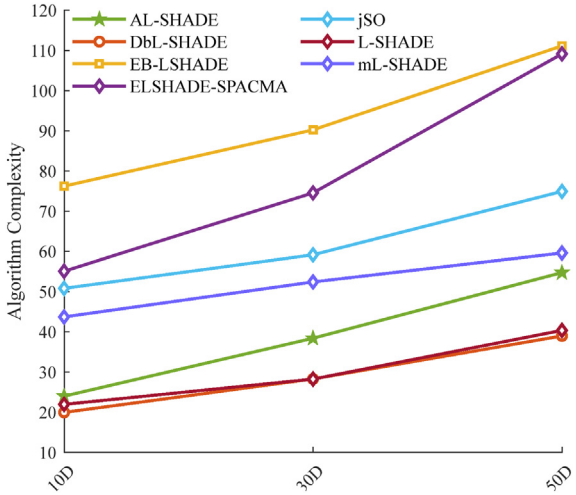

Fig. 7. Algorithm complexity comparison of seven algorithms.

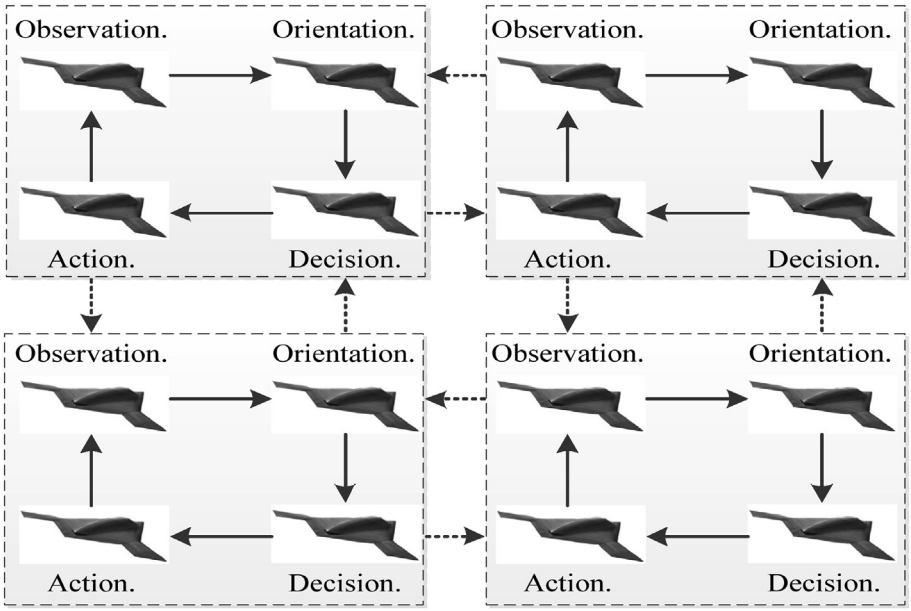

Fig. 8. The OODA combat theory in the UAV swarm operations.

5.1. Mathematical model description

At present, there are few publicly published literatures on the resource configuration of UAV swarm. Thus, a simple UAV swarm resource configuration method oriented to operation mission requirements based on the OODA combat theory is proposed in this section. This method is used to optimally configure UAV swarm resources. In the paper, the heterogeneous UAV swarm is denoted as $US = \{U_{Ob}, U_{Or}, U_D, U_A\}$, in which $U_{Ob} = \{U_{Ob}^1, U_{Ob}^2, \cdots, U_{Ob}^{n_1}\}$ is a cluster of observation UAVs, $U_{Or} = \{U_{Or}^1, U_{Or}^2, \cdots, U_{Or}^{n_2}\}$ is a cluster of orientation UAVs, $U_D = \{U_D^1, U_D^2, \cdots, U_D^{n_3}\}$ is a cluster of decision UAVs, and $U_A = \{U_A^1, U_A^2, \cdots, U_A^{n_4}\}$ is a cluster of action UAVs. It is noted there are differences in performance between UAVs of the same type to make full use of the advantages of coordinated operations. The targets cluster can be expressed as $T = \{T_1, T_2, \cdots, T_{N_T}\}$.

Since the low-cost, single-function UAV in the UAV swarm carry limited mission loads, one target may require multiple UAVs of the same type to perform corresponding tasks. Moreover, a UAV also could perform multiple targets in an operation mission. The goal of resource configuration is to configure UAVs of four types in the most efficient quantity to complete the corresponding operation mission before the UAV swarm performs the mission. In addition, the values of the UAVs included in the UAV swarm are also considered in the model. In summary, the resource configuration model of UAV swarm proposed in this paper is as follow.

$$
\begin{aligned}
\min \quad & f = \sum_{t=1}^{4} \Delta R_t + \sum_{i=1}^{N_U} (\min\{\sum_{j=1}^{N_T} x_{i,j}, 1\} \cdot V_i) + \delta \cdot Pu \\
& \Delta R_1 = \sum_{j=1}^{N_T} \sum_{i=1}^{N_{U1}} \left| x_{i,j} \cdot C_{Ob}^i - R_{Ob}^j \right| \\
& \Delta R_2 = \sum_{j=1}^{N_T} \sum_{i=1}^{N_{U2}} \left| x_{i,j} \cdot C_{Or}^i - R_{Or}^j \right| \\
& \Delta R_3 = \sum_{j=1}^{N_T} \sum_{i=1}^{N_{U3}} \left| x_{i,j} \cdot C_{D}^i - R_{D}^j \right| \\
& \Delta R_4 = \sum_{j=1}^{N_T} \sum_{i=1}^{N_{U4}} \left| x_{i,j} \cdot C_{A}^i - R_{A}^j \right|
\end{aligned} \tag{23}
$$

$$
\begin{aligned}
S.t. \quad & \sum_{j=1}^{N_T} x_{i,j} \leqslant Load_i, \qquad i = 1, 2, \cdots, N_U \\
& \begin{cases} \sum_{i=1}^{N_{U1}} x_{i,j} \cdot C_{Ob}^i \geqslant R_{Ob}^j \\ \sum_{i=1}^{N_{U2}} x_{i,j} \cdot C_{Or}^i \geqslant R_{Or}^j \\ \sum_{i=1}^{N_{U3}} x_{i,j} \cdot C_{D}^i \geqslant R_{D}^j \\ \sum_{i=1}^{N_{U4}} x_{i,j} \cdot C_{A}^i \geqslant R_{A}^j \end{cases}, \quad j = 1, 2, \cdots, N_T
\end{aligned} \tag{24}
$$

where V_i is the value of the i^{th} UAV, δ is the penalty function scale factor, and $Pu = 0$ if the configuration results meet the constraints in Eq. (24), otherwise, $Pu = 1$. $Load_i$ is the maximum number of target performed by the i^{th} UAV in one operation mission. $R_{Ob} = \{R_{Ob}^1, R_{Ob}^2, \cdots, R_{Ob}^{N_T}\}, R_{Or} = \{R_{Or}^1, R_{Or}^2, \cdots, R_{Or}^{N_T}\}, R_D = \{R_D^1, R_D^2, \cdots, R_D^{N_T}\}$ and $R_A = \{R_A^1, R_A^2, \cdots, R_A^{N_T}\}$ are the observation, orientation, decision, and action resources required for the target included in the operation mission. N_{U1}, N_{U2}, N_{U3}, and N_{U4} are the numbers of observation, orientation, decision, and action UAV in the UAV candidate pool from which UAV is selected to form a UAV swarm to perform the operation mission, and $N_U = N_{U1} + N_{U2} + N_{U3} + N_{U4}$ is the total number of UAV in the UAV candidate pool. $x_{i,j} \in \{0, 1\}$ is the decision variable, $x_{i,j} = 1$ indicates that the i^{th} UAV would perform the j^{th} target, otherwise$x_{i,j} = 0$. $C_{Ob}^i, C_{Or}^i, C_D^i$ and C_A^i are the observation, orientation, decision, and action capability of the i^{th} UAV, respectively.

5.2. Resource configuration coding

The UAV swarm resource configuration problem is a typical mixed integer programming problem, and the decision variable is discrete in the model built in Section 5.1. Thus, in order to utilize AL-SHADE to solve the UAV swarm resource configuration model proposed in this section, the mapping relationship between the individual position and the UAV swarm resource configuration result is established based on the coding method of real vector. The dimension D of the problem is $N_T \cdot (N_{U1} + N_{U2} + N_{U3} + N_{U4})$, and the corresponding relationship between the elements of the search individual position and the resource configuration decision variables is shown in Fig. 9. $d1 = N_T \cdot N_{U1}$,$d2 = N_T \cdot (N_{U1} + N_{U2})$, and

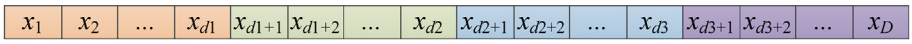

Fig. 9. The coding of UAV swarm resource configuration model.

$d3 = N_T \cdot (N_{U1} + N_{U2} + N_{U3})$. The search space of the solution is (0, 1). Moreover, the elements of the search individual position are rounded to the nearest integers due to the decision variable $x_{i,j}$ is 0 or 1.

5.3. Simulation experiments

5.3.1. Simulation condition

In the UAV swarm operation scenario, it is assumed that there are 24 UAVs in the UAV candidate pool that are listed in Table 7, and there are 4 targets to be performed which are listed in Table 8. Thus, the parameters setting of the UAV swarm resource configuration model in this paper is listed in Table 9. Note that normalized numbers are used to characterize capabilities of UAV and the resource requirements for performing operation target in order to simplify the UAV swarm operations resource configuration model. For the simplicity and readability of the paper, only four algorithms with the best performance in the results of section 4.3, i.e., AL-SHADE, EB-LSHADE, ELSHADE-SPACMA, and jSO are used to solve the model. The parameter settings of the four algorithms are consistent with section 4.1 and *MaxFEs* is set to 300,000 for all algorithms in this section. Moreover, the dimension of the problem is 96 that is regarded as a real-world constrained high-dimensional optimization problem.

5.4. Results and analysis

The statistical results of the four algorithms for solving the problem independently 30 times are shown in Table 10. AL-SHADE obtains the best result in terms of the statistical values 'Mean', 'Best', 'Worst', and 'Std' of the objective function among four algorithms, which shows that the average optimization performance and robustness of AL-SHADE on the constrained optimization is better than that of EB-LSHADE, ELSHADE-SPACMA, and jSO.

Moreover, the optimal UAV swarm resource configuration results derived from four algorithms for the resource requirements of the combat mission are shown in Table 11. According to the results obtained by AL-SHADE, there are three observation UAVs, three orientation UAVs, three decision UAVs, and three action UAVs in the UAV swarm, i.e., $U_{Ob} = \{U2, U3, U4\}, U_{Or} = \{U8, U9, U10\}, U_D = \{U13, U16, U18\}$ *and* $U_A = \{U19, U20, U21\}$.

To illustrate the convergence feature of four algorithms for constrained high-dimensional optimization, the mean convergence graphs based on 30 independent runs of four algorithms are shown as Fig. 10. AL-SHADE have similar convergence performance to EB-LSHADE. And AL-SHADE converge faster than ELSHADE-SPACMA but slower than jSO. Thus, we can conclude that AL-SHADE also provides a competitive performance in UAV swarm resource configuration problem that is also regarded as a constrained high-dimensional problem.

Table 7
UAV candidate pool for resource configuration.

No.	Type	Capability	Value	No.	Type	Capability	Value	No.	Type	Capability	Value	No.	Type	Capability	Value
U_1	Observation	0.1	0.2	U_7	Orientation	0.1	0.2	U_{13}	Decision	0.1	0.2	U_{19}	Action	0.2	0.2
U_2		0.2	0.3	U_8		0.2	0.3	U_{14}		0.2	0.3	U_{20}		0.3	0.3
U_3		0.3	0.4	U_9		0.3	0.4	U_{15}		0.3	0.4	U_{21}		0.4	0.4
U_4		0.4	0.5	U_{10}		0.4	0.5	U_{16}		0.4	0.5	U_{22}		0.5	0.5
U_5		0.5	0.6	U_{11}		0.5	0.6	U_{17}		0.5	0.6	U_{23}		0.6	0.6
U_6		0.6	0.7	U_{12}		0.6	0.7	U_{18}		0.6	0.7	U_{24}		0.7	0.7

Table 8
Resource requirements for performing operation missions.

No.	Resource requirements			
	Observation	Orientation	Decision	Action
T_1	0.4	0.3	0.5	0.4
T_2	0.5	0.4	0.5	0.5
T_3	0.6	0.5	0.6	0.6
T_4	0.7	0.6	0.6	0.7

Table 9
Parameters setting of the UAV swarm resource configuration model.

Parameter	Value	Parameter	Value
N_T	4	δ	10
N_{U1}	6	$Load_{Ob}$	3
N_{U2}	6	$Load_{Or}$	3
N_{U3}	6	$Load_D$	3
N_{U4}	6	$Load_A$	3

Table 10
Statistical results of four algorithms for resource configuration.

Algorithms	Mean	Best	Worst	Std	Time(s)
AL-SHADE	**5.85**	**4.80**	**6.60**	**0.37**	**5.23**
EB-LSHADE	5.93	5.20	6.70	0.40	6.35
ELSHADE-SPACMA	5.90	5.10	6.90	0.45	7.11
jSO	5.91	5.00	6.80	0.40	12.57

Table 11
The best result of the resource configuration derived from four algorithms.

Algorithms	No.	UAV in swarm			
		Observation	Orientation	Decision	Action
AL-SHADE	T_1	U_4	U_9	U_{13}, U_{16}	U_{21}
	T_2	U_2, U_3	U_{10}	U_{18}	U_{19}, U_{20}
	T_3	U_2, U_4	U_8, U_9	U_{18}	U_{19}, U_{21}
	T_4	U_3, U_4	U_8, U_{10}	U_{18}	U_{20}, U_{21}
EB-LSHADE	T_1	U_4	U_9	U_{17}	U_{22}
	T_2	U_2, U_3	U_{10}	U_{17}	U_{22}
	T_3	U_2, U_4	U_9, U_{10}	U_{14}, U_{16}	U_{23}
	T_4	U_3, U_4	U_9, U_{10}	U_{14}, U_{16}	U_{19}, U_{22}
ELSHADE-SPACMA	T_1	U_1, U_3	U_9	U_{17}	U_{21}
	T_2	U_2, U_3	U_{10}	U_{17}	U_{22}
	T_3	U_1, U_5	U_7, U_{10}	U_{18}	U_{19}, U_{21}
	T_4	U_2, U_5	U_9, U_{10}	U_{18}	U_{19}, U_{22}
jSO	T_1	U_5	U_9	U_{17}	U_{21}
	T_2	U_5	U_{10}	U_{17}	U_{19}, U_{20}
	T_3	U_1, U_5	U_7, U_{10}	U_{18}	U_{19}, U_{21}
	T_4	U_1, U_6	U_9, U_{10}	U_{18}	U_{20}, U_{21}

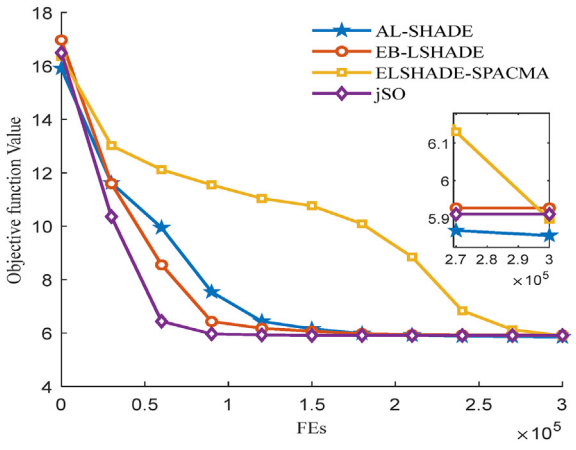

Fig. 10. Convergence graphs of four algorithms.

6. Conclusion and future work

In the study, a novel variant of L-SHADE, called AL-SHADE, is proposed to further improve its performance. The exploitation ability as well as the tuning exploitation and exploration ability have been improved according to the experimental statistics based on CEC 2018 test suite and CEC 2014 test suite. AL-SHADE and six competitors show different performance on CEC 2018 and CEC 2014 test suite. Particularly, AL-SHADE outperforms other competitors including DbL-SHADE, EB-LSHADE, ELSHADE-SPACMA, jSO, L-SHADE, and mL-SHADE. Moreover, AL-SHADE also showed excellent performance when solving the UAV swarm resource configuration problem described as a real-world constrained high-dimensional optimization problem.

In the future work, AL-SHADE parameter settings would be further adjusted according to the problem to obtain the optimal performance, and the UAV swarm resource configuration model is simplified to a certain extent compared with the actual situation that needs to be further adjusted to meet the actual combat requirements.

CRediT authorship contribution statement

Yintong Li: Conceptualization, Methodology, Data curation, Writing – original draft. **Tong Han:** Supervision, Project administration. **Huan Zhou:** Software, Validation. **Shangqin Tang:** Investigation, Writing-review & editing. **Hui Zhao:** Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China [grant numbers 62101590]; the Natural Science Foundation of Shaanxi Province [grant numbers 2021JM-223, 2021JM-224].

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ins.2022.05.058.

References

[1] G.-G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, Memetic Comput. 10 (2018) 151–164, https://doi.org/10.1007/s12293-016-0212-3.
[2] R. Tanabe, A. Fukunaga, Evaluating the performance of SHADE on CEC 2013 benchmark problems, in: 2013 IEEE Congr Evol. Comput. CEC 2013, 2013, pp. 1952–1959, https://doi.org/10.1109/CEC.2013.6557798.
[3] R. Storn, K. Price, Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces, 1995.
[4] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: Proc. 2014 IEEE Congr. Evol. Comput. CEC 2014. (2014) 1658–1665. https://doi.org/10.1109/CEC.2014.6900380.
[5] P.P. Biswas, P.N. Suganthan, G.A.J. Amaratunga, Optimal placement of wind turbines in a windfarm using L-SHADE algorithm, in: 2017 IEEE Congr Evol. Comput. CEC 2017 - Proc., 2017, pp. 83–88, https://doi.org/10.1109/CEC.2017.7969299.
[6] P.P. Biswas, P.N. Suganthan, G.A.J. Amaratunga, Minimizing harmonic distortion in power system with optimal design of hybrid active power filter using differential evolution, Appl. Soft Comput. 61 (2017) 486–496, https://doi.org/10.1016/j.asoc.2017.08.031.
[7] S.K. Goudos, G.V. Tsoulos, G. Athanasiadou, M.C. Batistatos, D. Zarbouti, K.E. Psannis, Artificial neural network optimal modeling and optimization of UAV measurements for mobile communications using the L-SHADE algorithm, IEEE Trans. Antennas Propag. 67 (2019) 4022–4031, https://doi.org/10.1109/TAP.2019.2905665.
[8] M. Hamdi, L. Idomghar, M. Chaoui, A. Kachouri, An improved adaptive differential evolution optimizer for non-convex Economic Dispatch Problems, Appl. Soft Comput. J. 85 (2019) 105868.
[9] K. Husen Khan, K. Bahadur Thapa, N. Raj Karki, Optimal coordination of directional overcurrent relays using enhanced L-SHADE algorithm, in: 2020 IEEE Int Conf. Power Syst. Technol. POWERCON 2020, 2020, https://doi.org/10.1109/POWERCON48463.2020.9230593.
[10] N. Awad, M.Z. Ali, R.G. Reynolds, A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization, in: 2015 IEEE Congr. Evol. Comput., IEEE, 2015, pp. 1098–1105, https://doi.org/10.1109/CEC.2015.7257012.
[11] S.M. Guo, J.S.H. Tsai, C.C. Yang, P.H. Hsu, A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set, in: 2015 IEEE Congr. Evol. Comput. CEC 2015 - Proc., 2015, pp. 1003–1010, https://doi.org/10.1109/CEC.2015.7256999.
[12] N.H. Awad, M.Z. Ali, P.N. Suganthan, R.G. Reynolds, An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems, in: 2016 IEEE Congr. Evol. Comput. CEC, 2016, pp. 2958–2965, https://doi.org/10.1109/CEC.2016.7744163.
[13] J. Brest, M.S. Maučec, B. Bošković, IL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization, in: IEEE Congr. Evol. Comput. CEC 2016, 2016, p. 2016, https://doi.org/10.1109/CEC.2016.7743922.
[14] A.W. Mohamed, A.A. Hadi, A.K. Mohamed, Differential evolution mutations: taxonomy, comparison and convergence analysis, IEEE Access 9 (2021) 68629–68662, https://doi.org/10.1109/ACCESS.2021.3077242.
[15] J. Brest, M.S. Maučec, B. Bošković, Single objective real-parameter optimization: Algorithm jSO, in: 2017 IEEE Congr Evol. Comput. CEC 2017 - Proc., 2017, pp. 1311–1318, https://doi.org/10.1109/CEC.2017.7969456.
[16] A.W. Mohamed, A.A. Hadi, A.M. Fattouh, K.M. Jambi, LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems, in: 2017 IEEE Congr Evol. Comput. CEC 2017 - Proc., 2017, pp. 145–152, https://doi.org/10.1109/CEC.2017.7969307.

[17] A.A. Hadi, A.W. Mohamed, K.M. Jambi, Single-objective real-parameter optimization: enhanced LSHADE-SPACMA algorithm, in: Stud. Comput. Intell., 2021: pp. 103–121. https://doi.org/10.1007/978-3-030-58930-1_7.
[18] A.W. Mohamed, A.K. Mohamed, Adaptive guided differential evolution algorithm with novel mutation for numerical optimization, Int. J. Mach. Learn. Cybern. 10 (2019) 253–277, https://doi.org/10.1007/s13042-017-0711-7.
[19] A. Viktorin, R. Senkerik, M. Pluhacek, T. Kadavy, L-SHADE algorithm with distance based parameter adaptation, Lect. Notes Electr. Eng. 465 (2018) 69–80, https://doi.org/10.1007/978-3-319-69814-4_7.
[20] A.P. Piotrowski, L-SHADE optimization algorithms with population-wide inertia, Inf. Sci. (Ny) 468 (2018) 117–141, https://doi.org/10.1016/j.ins.2018.08.030.
[21] J.F. Yeh, T.Y. Chen, T.C. Chiang, Modified L-SHADE for single objective real-parameter optimization, in: 2019 IEEE Congr. Evol. Comput. CEC 2019 - Proc., 2019, pp. 381–386, https://doi.org/10.1109/CEC.2019.8789991.
[22] A.W. Mohamed, A.A. Hadi, K.M. Jambi, Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization, Swarm Evol. Comput. 50 (2019), https://doi.org/10.1016/j.swevo.2018.10.006 100455.
[23] Y.C. Jou, S.Y. Wang, J.F. Yeh, T.C. Chiang, Multi-population modified L-SHADE for single objective bound constrained optimization, in: 2020 IEEE Congr. Evol. Comput. CEC 2020 - Conf. Proc. (2020). https://doi.org/10.1109/CEC48606.2020.9185735.
[24] X. Wang, C. Li, J. Zhu, Q. Meng, L-SHADE-E: Ensemble of two differential evolution algorithms originating from L-SHADE, Inf. Sci. (Ny) 552 (2021) 201–219, https://doi.org/10.1016/j.ins.2020.11.055.
[25] T.J. Choi, C.W. Ahn, An improved LSHADE-RSP algorithm with the Cauchy perturbation: iLSHADE-RSP, Knowl. Based Syst. 215 (2021) 106628.
[26] Z. Meng, C. Yang, Hip-DE: Historical population based mutation strategy in differential evolution with parameter adaptive mechanism, Inf. Sci. (Ny) 562 (2021) 44–77, https://doi.org/10.1016/j.ins.2021.01.031.
[27] Z. Tan, K. Li, Y. Wang, Differential evolution with adaptive mutation strategy based on fitness landscape analysis, Inf. Sci. (Ny) 549 (2021) 142–163, https://doi.org/10.1016/j.ins.2020.11.023.
[28] V. Stanovov, S. Akhmedova, E. Semenkin, Biased parameter adaptation in differential evolution, Inf. Sci. (Ny) 566 (2021) 215–238, https://doi.org/10.1016/j.ins.2021.03.016.
[29] S.X. Zhang, W.S. Chan, K.S. Tang, S.Y. Zheng, Adaptive strategy in differential evolution via explicit exploitation and exploration controls, Appl. Soft Comput. 107 (2021), https://doi.org/10.1016/j.asoc.2021.107494 107494.
[30] N.H. Awad, M.Z. Ali, J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 special session on single objective real-parameter numerical optimization, 2016.
[31] J.J. Liang, B.Y. Qu, P. N. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, 2014.
[32] R.L. Iman, J.M. Davenport, Approximations of the critical region of the fbietkan statistic, Commun. Stat. - Theory Methods 9 (1980) 571–595, https://doi.org/10.1080/03610928008827904.
[33] X. Zhou, W. Wang, T. Wang, X. Li, Z. Li, A research framework on mission planning of the UAV swarm, in: 2017 12th Syst. Syst. Eng. Conf., IEEE, 2017, pp. 1–6, https://doi.org/10.1109/SYSOSE.2017.7994984.
[34] L. Weng, Q. Liu, M. Xia, Y.D. Song, Immune network-based swarm intelligence and its application to unmanned aerial vehicle (UAV) swarm coordination, Neurocomputing 125 (2014) 134–141.
[35] Y. Spyridis, T. Lagkas, P. Sarigiannidis, J. Zhang, Modelling and simulation of a new cooperative algorithm for UAV swarm coordination in mobile RF target tracking, Simul. Model. Pract. Theory. 107 (2021) 102232, https://doi.org/10.1016/j.simpat.2020.102232.
[36] Z. Zhen, Y. Chen, L. Wen, B. Han, An intelligent cooperative mission planning scheme of UAV swarm in uncertain dynamic environment, Aerosp. Sci. Technol. 100 (2020) 105826, https://doi.org/10.1016/j.ast.2020.105826.
[37] X. Zhou, W. Wang, T. Wang, X. Li, T. Jing, Y. Deng, Continuous patrolling in uncertain environment with the UAV swarm, PLoS One 13 (8) (2018) e0202328, https://doi.org/10.1371/journal.pone.0202328.
[38] D. Xing, Z. Zhen, H. Gong, Offense–defense confrontation decision making for dynamic UAV swarm versus UAV swarm, Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng. 233 (15) (2019) 5689–5702.
[39] Y. Liu, H. Liu, Y. Tian, C. Sun, Reinforcement learning based two-level control framework of UAV swarm for cooperative persistent surveillance in an unknown urban area, Aerosp. Sci. Technol. 98 (2020) 1–20, https://doi.org/10.1016/j.ast.2019.105671.
[40] S. Gade, A. Joshi, Heterogeneous UAV swarm system for target search in adversarial environment, in: 2013 Int. Conf. Control Commun. Comput. ICCC 2013, 2013, pp. 358–363, https://doi.org/10.1109/ICCC.2013.6731679.
[41] Y. Lu, Y. Ma, J. Wang, L. Han, Task assignment of uav swarm based on wolf pack algorithm, Appl. Sci. 10 (2020) 1–17, https://doi.org/10.3390/app10238335.
[42] X. Fu, P. Feng, X. Gao, Swarm UAVs task and resource dynamic assignment algorithm based on task sequence mechanism, IEEE Access 7 (2019) 41090–41100, https://doi.org/10.1109/ACCESS.2019.2907544.
[43] D.J. Bryant, Rethinking OODA: toward a modern cognitive framework of command decision making, Mil. Psychol. 18 (2006) 183–206, https://doi.org/10.1207/s15327876mp1803_1.