

Enhanced LSHADE-SPACMA with fitness-directed selection pressure mutation and elastic archive for UAV trajectory planning and PV parameter identification



Wenhai Mao^{a,b}, Shengwei Fu^{a,b}, Guozhang Zhang^{a,b}, Haisong Huang^{a,b,*}, Guangming Gong^{a,b}, Yulu Liu^{a,b}, Honghai Fu^{a,b}, Yinwei Li^{a,b}, Yongpeng Zhao^{a,b}, Langlang Zhang^{a,b}, Jiawei Wang^{a,b}

^a Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, Guiyang 550025, China

^b Guizhou Equipment Manufacturing Digital Workshop Modeling and Simulation Engineering Research Center, Guiyang 550025, China

ARTICLE INFO

Keywords:

Differential evolution
Novel mutation strategy
External archive mechanism
Numerical optimization
UAV trajectory planning
Parameter identification

ABSTRACT

This paper presents a novel adaptive algorithm, ECLSHADE-SPACMA, designed to significantly improve the performance of its predecessor, LSHADE-SPACMA. The proposed algorithm incorporates four key innovations: First, it introduces a dynamic fitness-directed selection pressure mutation strategy to enhance the guidance of evolutionary direction. Second, it employs a nonlinear population reduction strategy based on an exponential function, enabling a smooth transition from global exploration to local convergence. Third, it proposes an archive pruning method based on elastic geometric weighting that effectively preserves high-quality solutions while maintaining population diversity. Finally, it establishes an adaptive probability-based local search mechanism guided by success rate feedback, thereby improving optimization efficiency in later stages. Experimental results show that the ECLSHADE-SPACMA algorithm not only significantly outperforms LSHADE-SPACMA on complex test problems but also surpasses many other high-performance optimizers. Especially when the problem dimension increases, its optimization performance advantage becomes more prominent. Further practical application evaluations demonstrate that ECLSHADE-SPACMA effectively overcomes the optimization challenges in unmanned aerial vehicle (UAV) trajectory planning within complex mountainous environments and achieves robust, efficient, and accurate performance in the parameter identification tasks of solar photovoltaic models.

1. Introduction

Evolutionary algorithms are now widely recognized as powerful tools for solving complex optimization problems, due to their gradient-free characteristics, strong global search abilities, and low sensitivity to problem formulations [1]. Storn and Price [2] proposed the differential evolution (DE) algorithm in 1997. It employs a differential mutation strategy to achieve a dynamic equilibrium between population diversity and search efficiency. This algorithm boasts core advantages such as simple parameter settings, strong robustness, and low computational complexity, leading to its widespread application in areas like UAV path planning [3], photovoltaic model parameter extraction [4], point cloud registration [5], image thresholding segmentation [6], communication

technology [7–9], and power system optimization [10].

Although the DE algorithm exhibits diverse advantages, it still encounters notable difficulties when applied to complex and dynamic optimization problems. The key challenges involve maintaining population diversity, avoiding premature convergence or stagnation, and balancing global exploration with local exploitation [11]. To overcome these limitations, researchers have proposed a range of algorithmic enhancements, including novel mutation strategies, external archive-based memory mechanisms, adaptive population sizing methods, and hybrid local search frameworks. These improvements aim to enhance the algorithm's capability to handle sophisticated optimization tasks while maintaining its computational efficiency and structural simplicity [12].

* Corresponding author.

E-mail addresses: gs.whmao24@gzu.edu.cn (W. Mao), gs.swfu22@gzu.edu.cn (S. Fu), gzzhang113@gmail.com (G. Zhang), hshuang@gzu.edu.cn (H. Huang), gs.gmgong22@gzu.edu.cn (G. Gong), gs.liyul24@gzu.edu.cn (Y. Liu), 18984669599@163.com (H. Fu), gs.ywli24@gzu.edu.cn (Y. Li), gs.ypzhao23@gzu.edu.cn (Y. Zhao), gs.langlangzhang23@gzu.edu.cn (L. Zhang), 15348606587@163.com (J. Wang).

For instance, in 2009, Zhang and Sanderson [13] proposed the JADE algorithm, which integrates an optional external archive with a parameter adaptation mechanism guided by feedback from successful solutions. This advancement significantly enhanced the algorithm's reliability and global exploration performance. In 2013, Tanabe et al. [14] introduced the success history-based adaptive differential evolution (SHADE) algorithm. Their approach improves the systematic selection of control parameters by leveraging configuration patterns that have demonstrated effectiveness in previous successful iterations. This refinement thereby further improved the algorithm's practical performance compared to JADE. To further enhance algorithmic performance, the LSHADE [15] framework was introduced in 2014, integrating historical memory mechanisms with linear population scaling. This integration effectively balances global exploration and local exploitation. In 2017, Mohamed et al. developed a hybrid optimization algorithm termed LSHADE-SPACMA [16], by combining L-SHADE, LSHADE-SPA [16], and CMA-ES [17]. An improved variant, ELSHADE-SPACMA [18], further strengthened the initial search capabilities through dynamic adaptation of parameter p and the incorporation of the directional mutation strategy AGDE [19]. Furthermore, Mohamed et al. [20] presented the first comprehensive taxonomy of differential evolution mutation strategies based on structural characteristics, providing practical guidelines to support the design and development of efficient optimization algorithms. To enhance exploitation effectiveness, Stannovov et al. [21] developed LSHADE-RSP in 2017 by incorporating a rank-driven selection pressure mechanism into the "current-to-pbest/1" mutation framework. Subsequently, Li and Han [22] proposed the IDE-EDA algorithm in 2022, which integrates LSHADE-RSP with the probabilistic modeling approach of estimation of distribution algorithms (EDA). This integration strategically combines the strong exploration capability of DE with the robust exploitation characteristics of EDA. In 2023, Li et al. introduced APSM-jSO [23] by incorporating adaptive parameter selection, FIFO archive updating, and a RSP-based mutation strategy into the jSO framework, thereby enhancing its exploitation capability. Subsequently, they developed MjSO [24], which utilized probabilistic selection and directional mutation mechanisms while eliminating the archive mechanism. This approach not only improved convergence performance but also effectively reduced computational complexity. In 2024, Zhang et al. [25] proposed the HAPI-DE algorithm, which integrates a mutation strategy based on hierarchical archives with the exploitation of promising information. Their approach significantly reduces the complexity of parameter selection and effectively addresses the inherent limitations of traditional mutation methods. Meanwhile, Meng et al. [26] proposed an algorithm named GOLDEN-DE, which integrates global opposition-based learning with diversity enhancement strategies and utilizes an exponential crossover mechanism to efficiently address continuous real-parameter optimization problems. Subsequently, Sun et al. [27] developed an integrated DE framework, termed fDE-ARM, based on an adaptive relay mechanism. This framework substantially improves solution accuracy and accelerates convergence performance. In 2025, Yi et al. [28] proposed the multi-population DE algorithm MPNBDE, which incorporates the B&D process and conditional opposition-based learning to enable automatic escape from local optima. Yuan et al. [29] introduced the PISCDE algorithm, integrating periodic intervention and strategic collaboration mechanisms to effectively mitigate premature convergence and stagnation issues. Zhu et al. [30] developed the SaUSDE algorithm by combining a mechanism for linearly increasing crossover probability with a four-strategy adaptive selection framework, aiming to achieve a balanced exploration-exploitation trade-off. Zhong et al. [31] designed the CFDE algorithm, which integrates DE/loser-to-best/loser-to-winner mutation strategies with competitive interaction and random initialization, thereby significantly enhancing its performance on complex optimization tasks. Chen et al. [32] proposed LSHADE-Code, a two-stage adaptive DE algorithm that combines complementary and ensemble mutation strategies to efficiently address high-dimensional optimization

problems.

The aforementioned DE variants enhance their performance through diverse approaches. Notably, the LSHADE-SPACMA algorithm achieves significant performance improvements within the fundamental DE framework. The LSHADE-SPACMA algorithm demonstrated remarkable performance in the global optimization competition 2017, obtaining superior results across a broad spectrum of benchmark optimization problems. However, when applied to highly complex or dynamically evolving optimization scenarios, this algorithm exhibits certain critical limitations, particularly regarding flexibility, adaptability, and local exploitation efficiency. Specifically:

- (1) Undirected differential vectors introduce excessive randomness into search trajectories, failing to effectively guide the evolutionary direction.
- (2) The population size decreases in a predetermined manner, impeding smooth transitions between different optimization stages.
- (3) The random replacement strategy employed in updating external archives may result in the premature removal of high-quality solutions, thereby compromising the algorithm's capacity to converge toward the global optimal solution.
- (4) The local exploitation mechanism in the later stages is relatively simplistic, with insufficient utilization of iteration information, thereby limiting convergence accuracy.

To address these limitations, this study introduces an enhanced algorithm named ECLSHADE-SPACMA. The primary objective of the proposed algorithm is to effectively integrate global exploration with thorough local exploitation capabilities, ensure dynamic and seamless transitions between optimization phases, and maintain both population diversity and solution quality throughout the search space. The key contributions of this work are summarized as follows:

- (1) A mutation strategy based on dynamic fitness-directed selection pressure is proposed with the aim of improving the guidance performance of the evolutionary direction.
- (2) A nonlinear population reduction strategy utilizing an exponential function is proposed to facilitate a smooth and adaptive transition from global exploration to local convergence.
- (3) An archive pruning method based on elastic geometric weighting is designed, effectively retaining high-quality solutions while preserving population diversity.
- (4) An adaptive probability local search mechanism guided by success rate feedback is constructed to substantially improve the algorithm's local optimization efficiency in the later stages of optimization.
- (5) Numerical optimization experiments were conducted using the CEC2014 and CEC2017 benchmark suites, and the results were systematically assessed in comparison with nine CEC competition-winning algorithms and four recently developed advanced variant methods.
- (6) The effectiveness of the ECLSHADE-SPACMA algorithm in UAV path planning for complex mountainous terrain was validated.
- (7) The effectiveness of the ECLSHADE-SPACMA algorithm was demonstrated through comprehensive experiments and comparisons addressing parameter identification problems in various solar photovoltaic models.

The remainder of this paper is organized as follows. Section 2 presents a comprehensive review of relevant literature. Section 3 provides a detailed exposition of the proposed ECLSHADE-SPACMA methodology. Section 4 describes the numerical experiments and their corresponding analyses. Section 5 establishes a mathematical model for addressing the three-dimensional path planning problem of UAVs in complex mountainous environments and proposes an effective solution strategy.

Section 6 evaluates the algorithm's performance in solving engineering optimization tasks through parameter identification experiments on solar photovoltaic systems. Finally, **Section 7** summarizes the key findings and outlines potential directions for future research.

2. Related work

2.1. DE

Differential evolution (DE) is a population-based evolutionary optimization algorithm that is widely employed to solve complex optimization problems. The core procedure of DE involves the iterative execution of mutation, crossover, and selection operations. This process enables guided exploration and gradual refinement of candidate solutions within the search space, thereby facilitating convergence toward the global optimum. Importantly, the dynamic design of the mutation strategy and the effective implementation of crossover play a critical role in determining the algorithm's optimization performance [33]. Additionally, the initial population P is generated through uniform sampling across the search space, as defined by Eq. (1).

$$x_i^j = L^j + \text{rand} \cdot (U^j - L^j) \quad (1)$$

where $\text{rand} \in (0, 1)$ denotes a uniformly distributed random variable, employed to modulate the stochastic properties of the parameters. The variable $i \in \{1, 2, \dots, NP\}$ indicates the index of individuals in the population, where NP denotes the population size; $j \in \{1, 2, \dots, D\}$ specifies the dimension index, and D represents the dimensionality of the optimization problem. Each dimension j of the optimization problem is constrained by corresponding boundary limits, where L^j and U^j respectively signify the lower and upper bounds of the j th-dimensional variable.

Mutation strategies generate mutation vectors v_i via vector difference mechanisms, and their classical forms encompass a variety of designs. Typical strategies include DE/rand/1, DE/best/1, DE/rand/2, DE/best/2, DE/current-to-best/1, and DE/current-to-rand/1, which are respectively expressed as Equations (2) through (7).

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) \quad (2)$$

$$v_i = x_{best} + F \cdot (x_{r1} - x_{r2}) \quad (3)$$

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) + F \cdot (x_{r4} - x_{r5}) \quad (4)$$

$$v_i = x_{best} + F \cdot (x_{r1} - x_{r2}) + F \cdot (x_{r3} - x_{r4}) \quad (5)$$

$$v_i = x_i + F \cdot (x_{best} - x_i) + F \cdot (x_{r1} - x_{r2}) \quad (6)$$

$$v_i = x_i + K \cdot (x_{r1} - x_i) + F \cdot (x_{r2} - x_{r3}) \quad (7)$$

Among these, variables x_{r1} through x_{r5} represent five distinct individuals selected via random sampling, while x_{best} denotes the current best solution identified during the iterative process. The scaling factor F controls the differential weight, and the parameter K is a randomly generated value within the interval $[0, 1]$. The variable v_i represents the position update quantity. If v_i exceeds the predefined boundary constraints, it will be adjusted in accordance with Eq. (8).

$$v_i^j = \begin{cases} (L^j + x_i^j)/2, & v_i^j < L^j \\ (U^j + x_i^j)/2, & v_i^j > U^j \end{cases} \quad (8)$$

The crossover operation selectively integrates the parameters of the mutant vector v_i and the original individual x_i on a dimension-by-dimension basis according to a predefined probability, thereby generating a trial vector. The logic underlying this operation is presented in Eq. (9).

$$u_i^j = \begin{cases} v_i^j, & \text{rand} < Cr \text{ or } j = j_{rand} \\ x_i^j, & \text{otherwise} \end{cases} \quad (9)$$

The crossover probability Cr determines the probability of fusion for each dimension parameter, while $j_{rand} \in \{1, 2, \dots, D\}$ represents a randomly generated dimension index.

Guided by the greedy strategy, the selection operation selects offspring individuals based on their fitness values and constructs a new population by integrating offspring individuals with lower fitness values with the elite individuals from the parent generation. The corresponding update rule is formally described in Eq. (10).

$$x_i = \begin{cases} u_i, & f(u_i) \leq f(x_i) \\ x_i, & \text{otherwise} \end{cases} \quad (10)$$

Here, $f(\cdot)$ denotes the objective function of the optimization problem under consideration.

2.2. LSHADE

The LSHADE [15] algorithm systematically enhances the parameter generation and search process of the classic differential evolution framework by incorporating historical memory mechanisms, dynamic parameter control, and a strategy for linearly reducing the population size.

During the parameter generation phase, the algorithm maintains two historical memory components, M_{CR} and M_F , which store information related to M_{CR} and M_F values that have shown superior performance in prior generations. These values are then employed to dynamically generate individual-specific CR_i and F_i parameters for each generation. The historical memory is configured with a fixed capacity of H , and all memory units are initialized to 0.5. For each individual x_i , its CR_i and F_i values are generated through the following procedure.

$$ri = \text{rand}(1, H) \quad (11)$$

$$CR_i = \begin{cases} 0, & M_{CR,ri} = \perp \\ \text{randn}(M_{CR,ri}, 0.1), & \text{otherwise} \end{cases} \quad (12)$$

$$F_i = \text{randc}(M_{F,ri}, 0.1) \quad (13)$$

Among them, randc represents sampling from the Cauchy distribution, while randn represents sampling from the normal distribution. Should a computed value for CR_i fall outside the defined interval $[0, 1]$, it shall be replaced with the nearest boundary value, either 0 or 1. In cases where F_i exceeds 1, F_i shall be set to 1. If F_i is less than or equal to 0, Eq. (13) shall be applied iteratively in order to obtain a valid value. According to Eq. (12), if $M_{CR,ri}$ has been assigned the terminal value \perp , then CR_i shall be set to 0.

In terms of the mutation strategy, LSHADE utilizes the "DE/current-to-pbest/1" approach to generate the mutation vector v_i , as presented in Eq. (14).

$$v_i = x_i + F_i \cdot (x_{pbest} - x_i) + F_i \cdot (x_{r1} - x_{r2}) \quad (14)$$

Among them, x_{pbest} signifies the top $p \times NP$ candidates systematically identified within the population, with parameter $p \in (0, 1)$ measuring the algorithm's greedy propensity—lower values reflect intensified greed. An individual x_{r1} is stochastically selected from population P , while x_{r2} originates from the merged set comprising P and external archive A . A is initially empty. Throughout the execution of the algorithm, individuals for which the performance of the parent vectors is inferior to that of their corresponding offspring vectors are incrementally incorporated into A . Once A reaches its predefined capacity, the existing members are randomly substituted with newly discarded parent individuals.

The update of historical memory relies on the parameter sets S_F and

S_{CR} of the successfully generated high-quality offspring, where S_F stores the effective scaling factor F_i values, and S_{CR} stores the effective crossover probability CR_i values. At the end of each generation, the memory units are updated using the weighted Lehmer mean.

$$M_{F,k} = \begin{cases} M_{F,k}, & S_F = \emptyset \\ \text{mean}_{WL}(S_F), & \text{otherwise} \end{cases} \quad (15)$$

$$M_{CR,k} = \begin{cases} M_{CR,k}, & S_{CR} = \emptyset \\ \perp, & \max(S_{CR}) = 0 \\ \text{mean}_{WL}(S_{CR}), & \text{otherwise} \end{cases} \quad (16)$$

The Equation for calculating the weighted Lehmer mean mean_{WL} is as follows:

$$\text{mean}_{WL}(S) = \frac{\sum_{n=1}^{|S|} \omega_n \cdot S_n^2}{\sum_{n=1}^{|S|} \omega_n \cdot S_n} \quad (17)$$

Here, S_n denotes either S_F or S_{CR} . The weight ω_n is determined by the fitness difference between the offspring individual \mathbf{u}_n and its corresponding parent individual \mathbf{x}_n , and the detailed calculation method is provided in Formula (18).

$$\omega_n = \frac{|f(\mathbf{u}_n) - f(\mathbf{x}_n)|}{\sum_{n=1}^{|S|} |f(\mathbf{u}_n) - f(\mathbf{x}_n)|} \quad (18)$$

The memory index k is incremented with each update and resets to 1 once it exceeds the capacity H , thereby ensuring that historical information is cyclically overwritten. It is particularly important to emphasize that when both S_F and S_{CR} are empty—indicating that no sub-vector demonstrates superior performance to the original vector—the values of M_F and M_{CR} remain unchanged.

The population size is dynamically adjusted via the linear population size reduction (LPSR) strategy.

$$NP_{g+1} = \text{round} \left[NP_{\text{init}} + \frac{NFE}{NFE_{\text{max}}} (NP_{\text{min}} - NP_{\text{init}}) \right], \quad (19)$$

Among them, round denotes rounding to the nearest integer. NP_{init} refers to the initial population size, and NP_{min} represents the minimum population size. NFE indicates the current number of evaluations, while NFE_{max} signifies the maximum number of evaluations. The algorithm operates iteratively, discarding individuals with inferior fitness in each generation, thereby progressively reducing the population size from the initial NP_{init} to NP_{min} . The algorithmic process terminates upon attaining the specified maximum limit of function evaluations.

2.3. LSHADE-RSP

The LSHADE-RSP [21] algorithm introduces three critical enhancements based on the LSHADE framework. Specifically, by incorporating sorting-based selection pressure, dynamic parameter constraints, and adaptive elite ratio adjustment, the search process is significantly refined. The advancement in the mutation strategy is embodied in the complete Equation for calculating the "DE/current-to-pbest-w/r" mutation vector \mathbf{v}_i , as presented below:

$$\mathbf{v}_i = \mathbf{x}_i + Fw_i \cdot (\mathbf{x}_{pbest} - \mathbf{x}_i) + F_i \cdot (\mathbf{x}_{pr1} - \mathbf{x}_{pr2}) \quad (20)$$

Among them, \mathbf{x}_{pr1} and \mathbf{x}_{pr2} are generated via non-uniform probability selection based on fitness ranking. Specifically, the population is first sorted in ascending order of fitness, and the ranking value of the individual at the i -th position is computed as follows:

$$Rank_i = k \times (NP - i) + 1 \quad (21)$$

Here, k represents the greedy control factor and is set to a value of 3. Additionally, NP denotes the current population size. The selection

probability of an individual is directly proportional to its ranking value, as expressed below:

$$pr_i = \frac{Rank_i}{\sum_{j=1}^{NP} Rank_j} \quad (22)$$

Dynamic weights Fw_i are adjusted according to the progression of the evolutionary stage.

$$Fw_i = \begin{cases} 0.7 \cdot F_i, & \text{if } NFE < 0.2 \cdot NFE_{\text{max}} \\ 0.8 \cdot F_i, & \text{if } 0.2 \cdot NFE_{\text{max}} \leq NFE < 0.4 \cdot NFE_{\text{max}} \\ 1.2 \cdot F_i, & \text{otherwise} \end{cases} \quad (23)$$

Secondly, following the generation of the scaling factor F_i and the crossover probability Cr_i via Eqs (13) and (12), respectively, F_i and Cr_i are dynamically adjusted throughout the evolutionary process in accordance with Eqs (24) and (25).

$$F_i = \begin{cases} \min(F_i, 0.7), & \text{if } NFE < 0.6 \cdot NFE_{\text{max}} \\ F_i, & \text{otherwise} \end{cases} \quad (24)$$

$$Cr_i = \begin{cases} \max(Cr_i, 0.7), & \text{if } NFE < 0.25 \cdot NFE_{\text{max}} \\ \max(Cr_i, 0.6), & \text{if } 0.25 \cdot NFE_{\text{max}} \leq NFE < 0.5 \cdot NFE_{\text{max}} \\ Cr_i, & \text{otherwise} \end{cases} \quad (25)$$

The proportion of elites, p , is transitioned from a fixed value to a dynamically adjusted parameter.

$$p = 0.085 + 0.085 \times \frac{NFE}{NFE_{\text{max}}} \quad (26)$$

The initial proportion, denoted as $p_{\text{min}} = 0.085$, is progressively increased to $p_{\text{max}} = 0.17$ throughout the evolutionary process. In addition, unlike LSHADE, LSHADE-RSP uniformly initializes all M_F values to 0.3 and all M_{CR} values to 0.8 during the parameter initialization phase. Furthermore, throughout the entire population evolution process, the values of $M_{F,H}$ and $M_{CR,H}$ are consistently maintained at 0.9.

2.4. LSHADE-SPACMA

To improve the computational efficiency of the algorithm, the LSHADE-SPACMA [16] algorithm incorporates the enhanced CMA-ES [17] optimizer into the LSHADE-SPA [16] framework, thereby establishing a coevolutionary architecture.

In the initial half of the function evaluation phase, LSHADE-SPA modifies the computational strategy for the F_i parameter within the LSHADE framework, thereby facilitating semi-parametric adaptive optimization. The corresponding formulation is presented below:

$$F_i = 0.45 + 0.1 \cdot \text{rand}, \quad NFE < NFE_{\text{max}}/2 \quad (27)$$

This Equation rigorously restricts F_i to the interval $[0.45, 0.55]$ in order to stabilize the early exploration behavior. The core mechanism of CMA-ES models the search space by utilizing a multivariate normal distribution.

$$\mathbf{x}_i = N(\mathbf{m}, \sigma^2 \mathbf{C}) \quad \forall i = 1 : n \quad (28)$$

The mean vector is updated through a weighted selection of the μ individuals with the best performance.

$$\mathbf{m} = \sum_{i=1}^{\mu} w_i \mathbf{x}_i \quad (29)$$

the weight w_i satisfies the following Equation:

$$\sum_{i=1}^{\mu} w_i = 1 \quad (30)$$

$$w_1 \geq w_2 \geq \dots \geq w_{\mu} \quad (31)$$

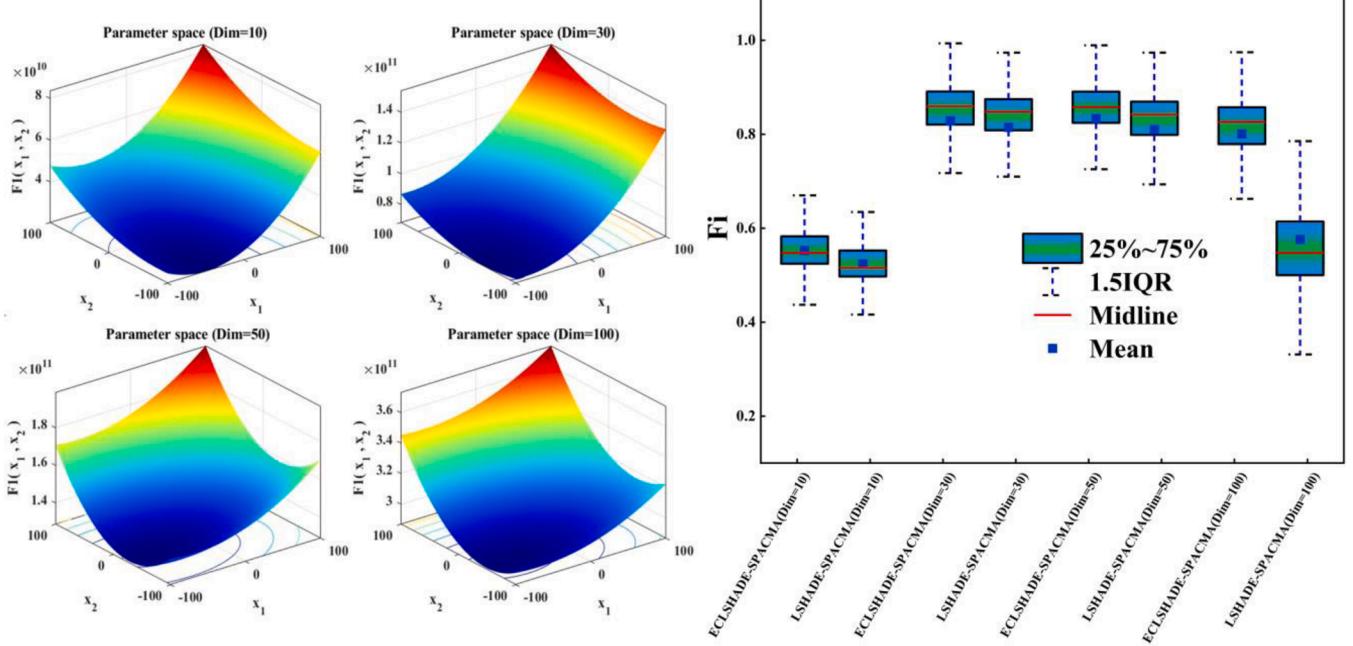


Fig. 1. Comparison of F_i distributions in LSHADE-SPACMA and ECLSHADE-SPACMA for the CEC2017 F1 function.

The key enhancement of the proposed approach lies in the incorporation of the crossover operation, as defined in [Equation \(9\)](#), following the generation of candidate solutions by CMA-ES. This significantly boosts the exploration capability of CMA-ES. Moreover, the covariance matrix σ and step size C are adaptively adjusted according to the evolutionary path.

The LSHADE-SPACMA framework commences with a shared population P , wherein each candidate solution x_i is capable of generating an offspring u by utilizing either the LSHADE strategy or the improved CMA-ES approach. The allocation of evolutionary strategies is controlled by the probability parameter FCP , which is randomly sampled from the memory module M_{FCP} . After each generation's evolutionary iteration, the M_{FCP} value is updated according to the relative performance evaluation of the two strategies, and this update is executed only when a superior offspring is successfully produced. This adaptive adjustment mechanism facilitates a progressive increase in the allocation of computational resources to the more effective evolutionary strategy. The core update mechanism is precisely defined by the following Equation:

$$M_{FCP,g+1} = (1 - c) \cdot M_{FCP,g} + c \cdot \Delta_{Alg1} \quad (32)$$

The memory update mechanism utilizes the learning rate parameter c as the primary control variable, which accurately defines the scale of parameter modifications. The algorithmic performance enhancement indicator Δ_{Alg1} , serving as a key evaluation metric, is calculated using [Eq. \(33\)](#).

$$\Delta_{Alg1} = \min(0.8, \max(0.2, \omega_{Alg1} / (\omega_{Alg1} + \omega_{Alg2}))) \quad (33)$$

Among them, the values 0.2 and 0.8 represent the lower and upper probability bounds assigned to each mutation operator. This configuration ensures concurrent execution of both algorithms, maintains algorithmic diversity, and achieves a balance between exploration and exploitation capabilities. Consequently, the FCP value is confined to the interval [0.2, 0.8]. Furthermore, ω_{Alg1} quantifies the cumulative difference between the updated and baseline fitness values in $Alg1$, serving as a performance metric to evaluate evolutionary progression within the algorithmic framework, as formulated in [Eq. \(34\)](#):

$$\omega_{Alg1} = \sum_{i=1}^n f(x_i) - f(u_i) \quad (34)$$

3. The proposed ECLSHADE-SPACMA method

3.1. Motivation

To address several critical limitations of the LSHADE-SPACMA algorithm in complex optimization tasks—specifically, the excessively dispersed search directions that impede effective guidance of the evolutionary process, the discontinuous transitions between algorithmic phases resulting from the linear population reduction scheme, the risk of losing high-quality solutions due to the stochastic replacement mechanism in the external archive, and the suboptimal convergence accuracy in later stages caused by overreliance on a single local exploitation strategy—this study proposes an enhanced variant termed ECLSHADE-SPACMA. The proposed algorithm represents a significant and independent contribution at both theoretical and structural levels, featuring a unified framework that goes beyond the simple aggregation of heuristic components or ad hoc parameter tuning. Unlike existing DE variants, ECLSHADE-SPACMA introduces a dynamic, feedback-driven architecture that explicitly captures the interplay among selection pressure, evolutionary direction, and adaptive local search, thereby strengthening the theoretical foundation of adaptive differential evolution.

Specifically, the approach integrates a novel dynamic fitness-directed selection pressure mutation strategy, implements a nonlinear population reduction method based on an exponential function, develops an elastically weighted geometric archive pruning mechanism, and incorporates an adaptive probability-based local search guided by success rate feedback. These comprehensively integrated enhancements collectively improve the algorithm's balance between exploration and exploitation, enhance convergence efficiency, and increase its robustness in addressing a wide range of complex optimization problems.

3.2. Dynamic fitness-directed selection pressure mutation

To address dispersed search directions in LSHADE-SPACMA under

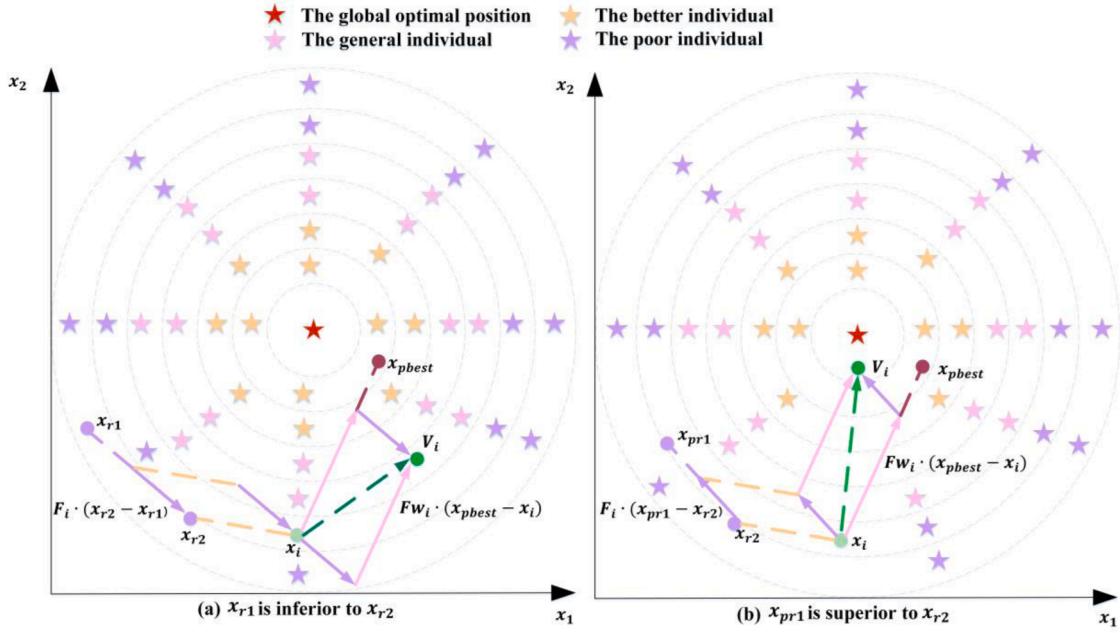


Fig. 2. A schematic illustration of dynamic fitness-directed selection pressure mutation within a two-dimensional search space.

complex optimization scenarios, we propose a dynamic fitness-directed selection pressure mutation strategy. First, we refine the scaling factor generation mechanism:

$$F_i = 0.50 + 0.05 \cdot \text{rand}, \quad \text{NFE} \leq 0.5 \cdot \text{NFE}_{\max} \quad (35)$$

In the context of full-dimensional global optimization for the F1 function within the CEC2017 benchmark suite, Fig. 1 provides a comparative analysis of the scaling factor distributions generated by the LSHADE-SPACMA algorithm according to Eq. (27) and its enhanced variant, ECLSHADE-SPACMA. The figure adopts a two-column layout: the left column displays the three-dimensional surface of the F1 function, with the horizontal axes representing the parameter space and the vertical axis indicating the objective value under full-dimensional global optimization; the right column presents the scaling factor distributions produced by both algorithms to facilitate direct comparison. Complete definitions of all CEC2014 and CEC2017 test functions employed in this study are detailed in Tables A.1 and A.2 of the supplementary material.

The results demonstrate that the F_i values achieved using the improved scaling factor generation mechanism are substantially higher, particularly in the 100-dimensional case, where the performance gap between the two algorithms is more evident. As a result, this enhancement strengthens the exploration capability of ECLSHADE-SPACMA during the global optimization process, thereby reducing the risk of premature convergence to local optima.

Building upon this foundation, we propose a mutation strategy that synergistically integrates rank-based selection pressure with fitness-directed differential guidance. The dual-mechanism framework enhances the mutation operation by incorporating phase-adaptive weight adjustment and fitness-directed vector control. Within this framework, the selection probability decreases in accordance with a predefined probability distribution as the fitness rank diminishes, thereby assigning higher selection priority to elite solutions. Meanwhile, real-time fitness evaluations dynamically modulate the difference vectors. The core mutation operator formalizes this cooperative mechanism as follows:

$$\mathbf{v}_i = \begin{cases} \mathbf{x}_i + Fw_i \cdot (\mathbf{x}_{pbest} - \mathbf{x}_i) + F_i \cdot (\mathbf{x}_{pr1} - \mathbf{x}_{r2}), & f(\mathbf{x}_{pr1}) \leq f(\mathbf{x}_{r2}) \\ \mathbf{x}_i + Fw_i \cdot (\mathbf{x}_{pbest} - \mathbf{x}_i) + F_i \cdot (\mathbf{x}_{r2} - \mathbf{x}_{pr1}), & f(\mathbf{x}_{pr1}) > f(\mathbf{x}_{r2}) \end{cases} \quad (36)$$

Among these, \mathbf{x}_{pr1} is selected from the current population P , which is sorted in ascending order of fitness through non-uniform probability

sampling, with its selection probability pr being defined by Eq. (22). \mathbf{x}_{r2} is randomly sampled from the union of population P and external archive A . The determination of the difference direction is directly guided by the fitness comparison: if $f(\mathbf{x}_{pr1}) \leq f(\mathbf{x}_{r2})$, the forward difference vector $\mathbf{x}_{pr1} - \mathbf{x}_{r2}$ is adopted; otherwise, the reverse difference vector $\mathbf{x}_{r2} - \mathbf{x}_{pr1}$ is utilized to mitigate random interference. The dynamic weight Fw_i is computed according to Eq. (23).

Within a two-dimensional search space, the graphical representation of the original "DE/current-to-pbest/1" mutation strategy is illustrated in Fig. 2(a), while the dynamic fitness-directed selection pressure mutation strategy is depicted in Fig. 2(b). As can be observed from these visualizations, the mutation vector \mathbf{v}_i generated using the dynamic fitness-directed selection pressure mutation approach demonstrates notable advantages over the mutation vector obtained through the original mutation strategy.

3.3. Nonlinear population reduction strategy utilizing an exponential function

Within differential evolution, conventional population reduction approaches often result in an imbalance between exploration and exploitation, primarily due to fixed decay rates or abrupt discrete-stage adjustments. Although the commonly used exponential decay technique mitigates sudden population changes through gradual transitions, it still encounters persistent limitations: a fixed *a priori* decay rate fails to accommodate the demands of multi-stage optimization, while complex dynamic control schemes introduce additional computational overhead. To address these limitations, this study proposes a nonlinear population reduction strategy based on an exponential function. The proposed approach employs an exponential formulation and achieves adaptive control of population size through a computationally efficient tuning mechanism, thereby effectively balancing smooth transitions with context-specific adaptability. The corresponding mathematical formulation is as follows:

$$NP(g+1) = \text{Round}\left[NP_{\text{init}} \cdot r^k \right] \quad (37)$$

$$r = \frac{NP_{\min}}{NP_{\text{init}}} \quad (38)$$

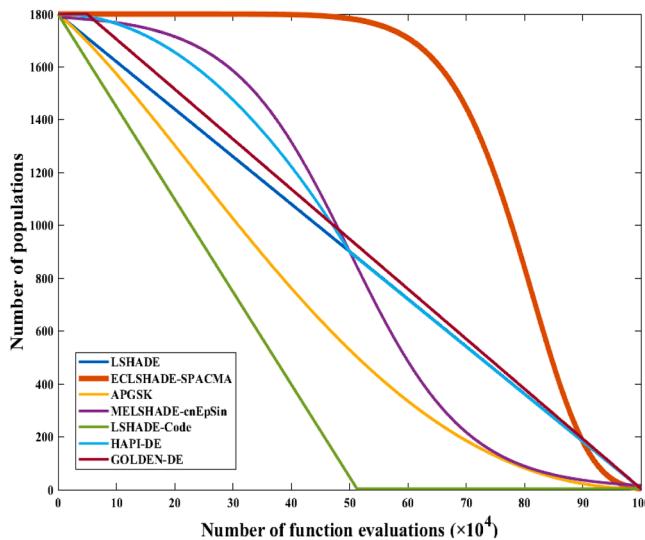


Fig. 3. Population reduction strategies curve comparison.

$$t = \frac{NFE}{NFE_{\max}} \quad (39)$$

k satisfies the following Equation:

$$r^k = (1 - c) + c \cdot r \quad (40)$$

Here, $NP(g+1)$ represents the population size of the $(g+1)$ -th generation, with c being set to 0.9. Sensitivity analysis for parameter c is detailed in Section 4.3. By predefining the curvature control factor c , the attenuation trajectory is dynamically reshaped in real time. When the evolutionary progress t equals c , the curve automatically adjusts itself to pass through the target point NP_{target} . The Equation for calculating the target point NP_{target} is presented as follows:

$$NP_{\text{target}} = (1 - c) \cdot NP_{\text{init}} + c \cdot NP_{\min} \quad (41)$$

To rigorously demonstrate the substantial advantages of the nonlinear population reduction strategy based on an exponential function over other population reduction approaches, a systematic comparative analysis was conducted by incorporating six classical and advanced strategies into the ECLSHADE-SPACMA framework. Comprehensive experimental details are presented in Section 4.4. These strategies encompass fundamental methods such as linear reduction [15] (LSHADE) and exponential reduction [34] (APGSK), as well as recently developed high-performance variants, including enhanced linear reduction [32] (LSHADE-Code), nonlinear Sigmoid-based reduction [35] (MELSHADE-cnEpSin), and specialized two-phase approaches [25, 26] (HAPI-DE and GOLDEN-DE). Under the constraints of a maximum population size of 1800 and a maximum function evaluation count of 1000000, Fig. 3 presents comparison curves for the seven population reduction strategies, illustrating their population size trajectories across function evaluations. The ECLSHADE-SPACMA algorithm, which employs the exponential nonlinear population reduction strategy, demonstrates a distinct advantage: maintaining a relatively large population size during the early stages enhances global exploration, while its subsequent efficient reduction accelerates local exploitation. In contrast, the benchmark strategies exhibit either premature convergence or insufficient exploration capability.

3.4. Elastic geometric weighted archive pruning strategy

Since Zhang and Sanderson [13] introduced the external archive technique into the Differential Evolution (DE) algorithm in 2009, it has been widely adopted across various DE variants and has significantly

enhanced algorithmic performance. Despite the numerous archive management strategies proposed to date, maintaining a balance between archive diversity and the retention of high-quality solutions remains a challenge. For example, the classical random replacement strategy, although beneficial for preserving population diversity, may result in the premature replacement of high-quality solutions, thereby undermining the algorithm's ability to converge toward the global optimum. Conversely, while the elite screening mechanism is effective in preserving superior solutions, its excessive emphasis on the optimal region may compromise diversity, thereby impairing the algorithm's global exploration capability. To address these limitations, this paper introduces the elastic geometric weighted archive pruning strategy, which combines geometric weighting with probabilistic screening mechanisms. The core procedure involves arranging individuals in ascending order based on their fitness values. When the archive exceeds its capacity limit, the proposed strategy sequentially calculates the retention probability weight W_i for each individual indexed as i .

$$W_i = base_k \cdot (1 + 2g) \cdot \exp(\alpha \cdot (A - i)) + 1 \quad (42)$$

$$g = \frac{NFE}{NFE_{\max}} \quad (43)$$

Among these parameters, the term $base_k$ denotes the basic weight coefficient, which is set to a value of 1.1. The variable g represents the evolution stage ratio. The parameter α governs the geometric ratio attenuation rate and determines the steepness of the weight distribution, with its value fixed at 0.05. Additionally, A signifies the current size of the archive. Once all weights W_i have been computed, the algorithm proceeds to normalize them into retention probabilities.

$$P_i = \frac{1/W_i}{\sum_{j=1}^A 1/W_j} \quad (44)$$

Subsequently, the retention status of individuals is determined via probabilistic sampling, and this procedure is iteratively executed until the archive size is reduced to the predefined upper limit. This mechanism not only ensures the diversity of the solution space and the preservation of elite individuals but also dynamically modulates the weight distribution according to the evolutionary stage ratio g , thereby achieving an adaptive equilibrium of selection pressure.

3.5. Adaptive probability local search mechanism guided by success rate feedback

To enhance local exploitation capability during the later stages of the ECLSHADE-SPACMA framework while maintaining computational efficiency, this study incorporates an adaptive probabilistic local search mechanism guided by success rate feedback. The local search procedure is activated when the cumulative function evaluations reach a pre-defined threshold (denoted as NFE_{LS}), set at 75% of the total evaluation budget. This carefully designed threshold ensures adequate global exploration during the early optimization phases while allowing focused local refinement in later stages. The formal activation criterion is mathematically formulated as follows:

$$\text{activate_LS} = \begin{cases} 1 & \text{if } NFE > 0.75 \times NFE_{\max} \text{ and } rand < P_{LS} \\ 0 & \text{otherwise} \end{cases} \quad (45)$$

When the activation conditions are met, the algorithm invokes the sequential quadratic programming (SQP) method and conducts $0.02 \times NFE_{\max}$ fitness evaluations per iteration to locally optimize the current best solution x_{best} , thereby generating an improved solution x_{new} . Following the completion of the search process, the algorithm proceeds to validate the feasibility of the enhanced solution.

$$f(x_{\text{new}}) < f(x_{\text{best}}) \quad (46)$$

Algorithm 1

ECLSHADE-SPACMA.

```

1:   G = 1, NG = Ninit, A = ∅, NFE = 0
2:   Initialize population PG
3:   Evaluate fitness of PG and update NFE
4:   Initialize memory MCR, MF, MFCP to 0.5
5:   Initialize CMA-ES parameters
6:   While NFE < NFEmax Do
7:     SCR = ∅, SF = ∅, SFCP = ∅
8:     For i = 1 to N Do
9:       ri = Random selection from [1, H]
10:      CRi,G = randni(MCR,i, 0.1)
11:      FCPi,G = MFCP,i
12:      If NFE < NFEmax/2
13:        Fi,G = 0.5 + 0.05 * rand
14:      Else
15:        Fi,G = randci(MF,i, 0.1)
16:      End
17:    End
18:    [PECLSHADE,G, PCMA-ES,G] = Split(PG, FCPG)
19:    VG,ECLSHADE = Generate trial vectors by Eq. (36)
20:    VG,CMA-ES = Generate trial vectors by CMA-ES
21:    UG = Generate trial vectors by Eq. (9) and Eq. (13)
22:    Calculate the fitness value of UG
23:    Update NFE
24:    Update PG by . (10)
25:    Store successful FCPG, FG, and CRG
26:    Update archive A
27:    If (archive size > |A|)
28:      The individuals are removed by elastic geometric weighted archive
pruning mechanism
29:    End
30:    Update memory MCR, MFCP
31:    If NFE > NFEmax/2
32:      Update memory MF
33:    End
34:    Update NG by . (37) and . (39)
35:    If NG < NG+1
36:      Sort individuals in P based on their fitness
37:      values and delete lowest NG - NG+1 members;
38:      Resize archive size |A| according to new |P|;
39:    End
40:    If NFE > NFEmax * 0.75 and rand(0, 1) < PLS
41:      Apply the SQP method to refine current best solution for up to
0.02*NFEmax fitness evaluations
42:      If fitness(optimized_solution) < fitness(xbest)
43:        Replace xworst with optimized_solution in PG
44:        Update success_count
45:      End
46:      Update total_count, NFE
47:      Update success_rate and PLS by Eq. (47) and Eq. (48)
48:    End
49:    Update CMA-ES parameters
50:    G=G+1
51:  End

```

If the inequality holds, the current local search is classified as an effective optimization, and a success event is logged; otherwise, it is treated as an unsuccessful attempt. When the local search is classified as effective, the elite replacement strategy is invoked, where x_{new} replaces x_{worst} , the individual with the poorest fitness in the population. Upon completion of the replacement, the local search success counter $ls_success_count$ is incremented by 1; otherwise, only the total local search trigger counter ls_total_count is incremented by 1. Regardless of the outcome, ls_total_count is incremented by 1 after each local search attempt to ensure that the trigger probability can be dynamically adjusted based on the cumulative success rate for the subsequent generation iteration. The historical success rate P_{LS} is computed using the following Equation:

$$success_rate = \frac{ls_success_count}{ls_total_count} \quad (47)$$

Subsequently, the trigger probability P_{LS} is adaptively updated via a linear feedback mechanism:

$$P_{LS} = P_{LSmin} + success_rate \times (P_{LSmax} - P_{LSmin}) \quad (48)$$

The minimum trigger probability is set to $P_{LSmin} = 0.01$, and the maximum trigger probability is set to $P_{LSmax} = 0.4$. This "trigger-search-feedback-regulation" closed-loop control structure allows the algorithm to adaptively adjust the intensity of local exploitation according to the optimization progress and search performance.

In conclusion, the pseudocode of ECLSHADE-SPACMA is outlined in [Algorithm 1](#).

4. Numerical experiments were carried out utilizing the CEC benchmark test suite

4.1. Comprehensive overview of the CEC test suite and experimental platform

This study evaluates the performance of ECLSHADE-SPACMA across 10-, 30-, 50-, and 100-dimensional problem settings using the IEEE CEC 2014 and 2017 [36,37] benchmark test suites. These well-established benchmark functions, maintained and updated by the IEEE Congress on Evolutionary Computation (CEC) Technical Committee, are based on publicly available datasets. According to the specifications of the CEC test suites, the search space is constrained within $[-100, 100]^D$. For both the CEC2014 and CEC2017 benchmark sets, the maximum number of function evaluations (NFE_{max}) is set to 10000*D, resulting in 100000, 300000, 500000, and 1000000 evaluations for dimensions D = 10, 30, 50, and 100, respectively. Moreover, comprehensive descriptions of the test functions employed in CEC14 and CEC17 are presented in Tables A.1 and A.2 of the supplementary materials. It is important to note that Function F2 in the CEC2017 suite was excluded due to documented numerical instability in high-dimensional scenarios. Unless otherwise specified, all numerical results were obtained from 51 independent algorithmic runs.

All experiments were carried out using MATLAB R2023a on a computer system equipped with an Intel(R) Core (TM) i9-13900K processor operating at 3.00 GHz and 64 GB of RAM.

4.2. Parameter configuration for the algorithm

The parameter configuration of the ECLSHADE-SPACMA algorithm plays a central role in this study. In the comparative experiments, its performance was evaluated against 13 advanced algorithms: LSHADE [15], EBOwithCMAR [38], LSHADE-SPACMA [16], HSES [39], ELSHADE-SPACMA [18], IMODE [40], AGSK [41], EA4eig [42], IDE-EDA [22], APSM-jSO [23], AL-SHADE [43], mL SHADE-RL [44], mL SHADE-SPACMA [45]. [Table 1](#) presents the parameter configurations for all competing algorithms. A consistent stopping criterion was employed across all experiments: the maximum number of function evaluations. The evaluation was conducted on the CEC2014 and CEC2017 benchmark suites at four standard dimensional levels: 10, 30, 50, and 100. To ensure statistical robustness, each algorithm was executed independently 51 times. Performance assessment was based on two non-parametric statistical methods: the Wilcoxon signed-rank test [46] and the Friedman mean rank test [47]. An essential component of the evaluation framework is the primary error metric, which measures the absolute difference between the obtained solution and the global optimum for each benchmark function.

4.3. Parametric sensitivity assessment

4.3.1. Sensitivity analysis of curvature in exponential-function-based nonlinear population reduction

This section presents a sensitivity analysis of the curvature parameter c in an exponential-function-based nonlinear population size reduction strategy. Through systematic evaluation of the influence of varying

Table 1

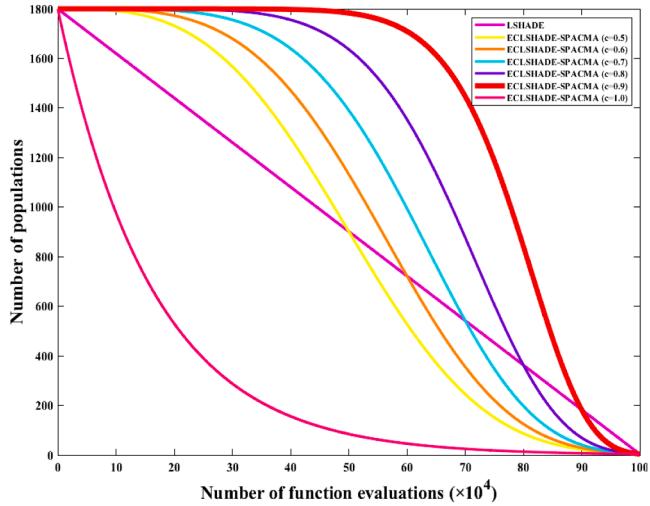
The specified algorithm parameter configurations.

Algorithm	Parameters configurations
L SHADE	$NP_{init} = 18 * D, NP_{min} = 4, p_best_rate = 0.11, A = 2.6, H = 6.$
EBOwithCMAR	For EBO, $PS_{1,max} = 18 * D, PS_{1,min} = 4, PS_{2,max} = 46.8 * D, PS_{2,min} = 10, H = 6.$ For CMAR, $PS_3 = 4 + (3\log(D)), \sigma = 0.3, CS = 100$ and 200 for the 10D and 30D, respectively, and 300 for 50D and 100D problems, For local search, $prob_{ls} = 0.1$ and $cfe_{ls} = 0.25 * FE_{max}.$
L SHADE-SPACMA	$NP_{init} = 18 * D, NP_{min} = 4, p_best_rate = 0.11, A = 1.4, H = 5, \sigma = 0.5, L_rate = 0.8.$
HSES	$\lambda = \text{round}(4 + 3 * \log(D)), C = 100, 200, \text{ and } 300 \text{ for the 10D, 30D, and 50D, respectively,}$ For local search, $prob_{ls} = 0.1$ and $cfe_{ls} = 0.25 * FE_{max}.$
EL SHADE-SPACMA	$NP_{init} = 18 * D, NP_{min} = 4, P_{max} = 0.3, P_{min} = 0.15, A = 1.4, H = 5, threshold = MaxFES/2, P_{AGDE} = 0.1, F_{CP} = 0.5, c = 0.8.$
IMODE	$NP_{init} = 6 * D^2, NP_{min} = 4, A = 2.6, H = 20 * D, FE_{LS} = 0.85 * FE_{max}.$
AGSK	$NP_{init} = 40 * D, KF = 0.5, KR = 0.9, Kw_P = [0.85, 0.05, 0.05, 0.05],$ The learning rate $c = 0.05.$
EA4eig	$NP_{init} = 100, NP_{min} = 10, p_best_rate = 0.11, A = 2.6, H = 4.$
IDE-EDA	$NP_{init} = 75 * D^{(2/3)}, NP_{min} = 4, P_{max} = 0.17, P_{min} = 0.085, k = 3, H = 5, r^{arc} = 1, \tau = 0.9.$
APSM-jSO	$NP_{init} = 75 * D^{(2/3)}, NP_{min} = 4, p_{max} = 0.17, p_{min} = 0.085, A = 1.3 * NP, H = 6, k = 3.$
AL-SHADE	$NP_{init} = 18 * D, NP_{min} = 4, p_best_rate = 0.11, A = 2.6, H = 6, e=0.5.$
mL SHADE-RL	$NP_{init} = 18 * D, NP_{min} = 4, p_best_rate = 0.11, A = 1.4, H = 5, P_c = 0.4, P_s = 0.5, P_{ls} = 0.01, FE_{LS} = 0.85 * FE_{max}.$
mL SHADE-SPACMA	$NP_{init} = 18 * D, NP_{min} = 4, p_best_rate = 0.11, A = 1.4, H = 5, \sigma = 0.5, L_rate = 0.8, p = 0.01, k = 3.$
ECL SHADE-SPACMA	$NP_{init} = 18 * D, NP_{min} = 4, p_best_rate = 0.11, A = 1.4, H = 5, \sigma = 0.5, Lrate = 0.8, c = 0.9, base_k = 1.1, \alpha = 0.05, NFE_{LS} = 0.75, P_{LSmin} = 0.01, P_{LSmax} = 0.4.$

Table 2

Detailed parameter settings for different cases.

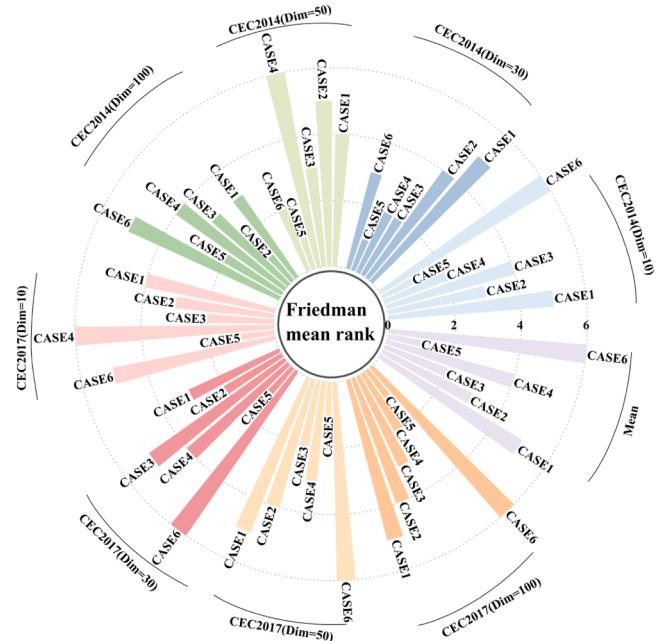
ID	Case1	Case2	Case3	Case4	Case5	Case6
c	0.5	0.6	0.7	0.8	0.9	1.0

**Fig. 4.** The population size reduction process governed by parameter c .

parameter values on algorithmic performance, the optimal threshold for enhancing algorithmic stability is determined and applied to the equation presented below.

$$r^k = (1 - c) + c \cdot r \quad (40)$$

This study designed six parameter configuration schemes (as detailed in Table 2) to conduct a systematic comparative analysis. Fig. 4 illustrates the dynamic evolution of population size with respect to the number of function evaluations under varying c parameter settings. Experimental results demonstrate that as the value of c increases, the decay rate of the population size decreases, thereby enhancing the preservation of global exploration capabilities. In contrast, when $c = 1.0$, a sharp decline in population size is observed, leading to inadequate global search performance. Statistical outcomes derived from the Friedman mean rank test (presented in Fig. 5) reveal that although Case 5 underperforms slightly compared to Case 2 in CEC2014 (Dim = 100), it achieves superior optimization performance across all other test

**Fig. 5.** Friedman mean rank results under different configurations of the parameter c .

environments, including different dimensional configurations of CEC2014 and the complete CEC2017 test suite. Notably, Case 5 attains the lowest average ranking, indicating its overall superiority. Therefore, under the ECSHADE-SPACMA framework developed in this research, Case 5 is identified as the most suitable parameter setup for c .

4.4. Sensitivity analysis of weighting parameters in the elastic geometric weighted archive pruning strategy

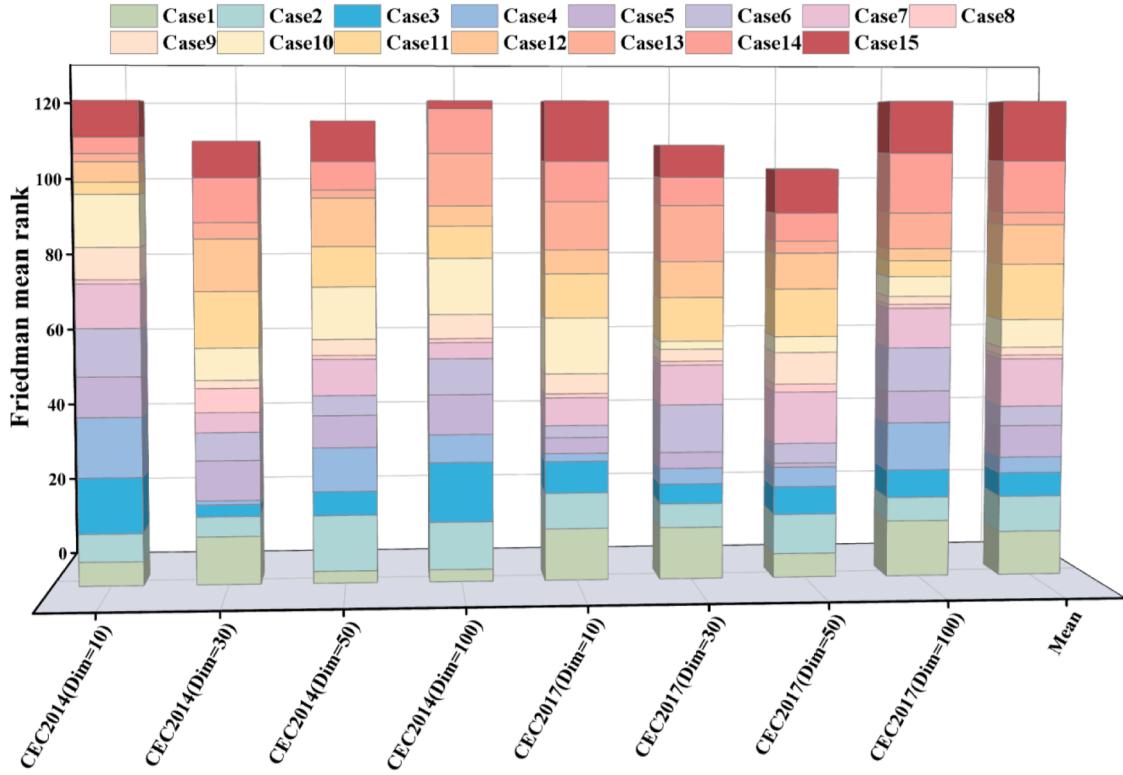
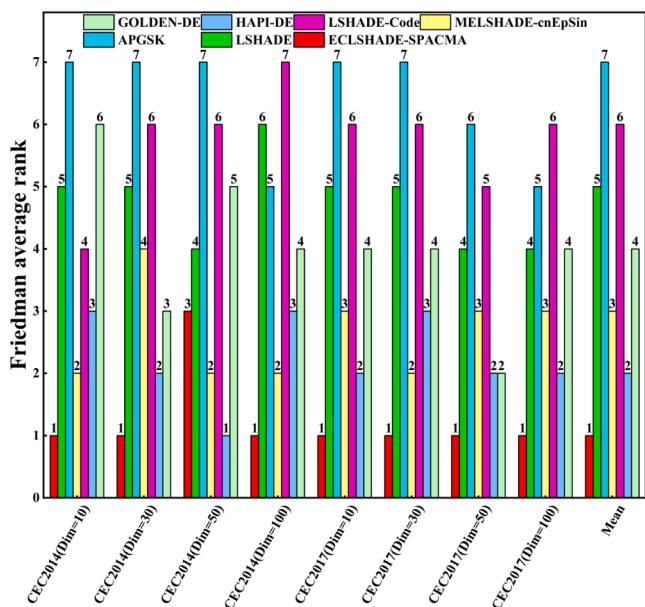
In this section, a systematic sensitivity analysis is conducted on two key weighting parameters in the elastic geometric weighted archive pruning strategy: the base weight coefficient ($base_k$) and the geometric attenuation parameter (α). The objective of this study is to assess the influence of these parameters on the robustness of the algorithm and to incorporate them into the equation presented below.

This study proposes 15 parameter configuration schemes (see Table 3 for details) and evaluates their performance using the Friedman mean rank test. The results are presented in Fig. 6. The analysis reveals

Table 3

Detailed parameter settings for different cases.

ID	Case1	Case2	Case3	Case4	Case5	Case6	Case7	Case8	Case9	Case10	Case11	Case12	Case13	Case14	Case15
$base_k$	0.9	0.9	0.9	1.0	1.0	1.0	1.1	1.1	1.1	1.2	1.2	1.2	1.3	1.3	1.0
α	0.04	0.05	0.06	0.04	0.05	0.06	0.04	0.05	0.06	0.04	0.05	0.06	0.04	0.05	0.06

**Fig. 6.** Friedman mean rank results under different configurations of the parameters $base_k$ and α .**Fig. 7.** Friedman mean rank comparison of population reduction strategies.

that, with the exception of CEC2014 (Dim = 30), where Case8 performs worse than Cases 4, 9, 3, 13, 2, and 7, and CEC2017 (Dim = 50), where it is outperformed by Case 5, Case8 achieves the best performance across all test suites and dimensions. Notably, in terms of average performance, Case8 also ranks first. Therefore, within the ECLSHADE-SPACMA framework, Case8 is selected as the optimal parameter configuration.

4.5. Comparative study on population reduction strategies

The primary objective of this section is to conduct a rigorous evaluation of the performance advantage provided by the proposed nonlinear population reduction strategy utilizing an exponential function compared to existing representative strategies. To achieve this, we systematically integrated the proposed strategy with six benchmark strategies—LSHADE, APGSK, LSHADE-Code, MELSHADE-cnEpSin, HAPI-DE, and GOLDEN-DE—into the ECLSHADE-SPACMA framework and performed a thorough comparative analysis across multiple standard test sets.

Fig. 7 displays the Friedman mean rank test results obtained from comparative experiments that assessed various population reduction strategies across different test sets. The findings reveal that while the ECLSHADE-SPACMA implementing the proposed strategy exhibited inferior performance relative to HAPI-DE and MELSHADE-cnEpSin on the CEC2014 (Dim=50) test set, it achieved superior performance across all other test sets, including additional dimensions of CEC2014 and the CEC2017 test set. Notably, its average Friedman mean rank was significantly lower than those of all competing strategies. Taken together,

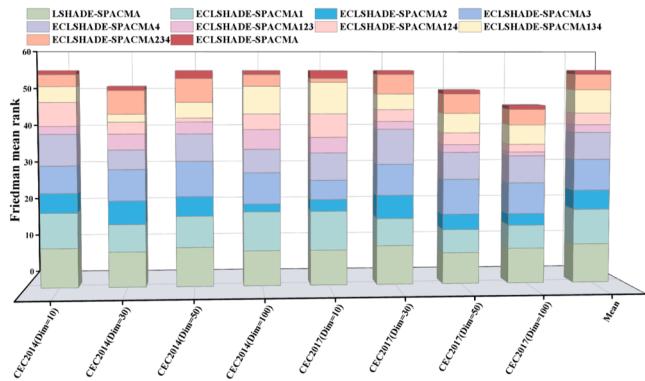


Fig. 8. Evaluation of the impacts of various enhancement strategies.

these experimental results confirm that the nonlinear population reduction strategy based on an exponential function consistently and significantly outperforms the other six benchmark strategies in terms of overall performance.

4.6. Ablation experiment

This study aims to improve the optimization performance of LSHADE-SPACMA by incorporating four enhanced strategies: a dynamic fitness-directed selection pressure mutation strategy, a nonlinear population reduction strategy utilizing an exponential function, an elastic geometric weighted archive pruning strategy, and a success rate feedback-guided adaptive probability local search mechanism. To systematically assess both the individual and synergistic contributions of these strategies, eight variants of ECLSHADE-SPACMA were developed—namely, ECLSHADE-SPACMA1 through ECLSHADE-SPACMA4, along with the combined variants ECLSHADE-SPACMA123, ECLSHADE-SPACMA124, ECLSHADE-SPACMA134, and ECLSHADE-SPACMA234. All variants were configured strictly in accordance with the parameter settings specified for LSHADE-SPACMA and ECLSHADE-SPACMA in Table 1.

The Friedman average rank test results presented in Fig. 8 indicate that the complete ECLSHADE-SPACMA algorithm, integrating all four enhancement strategies, achieves superior performance across the majority of benchmark test suites and dimensional settings. It is marginally outperformed only by ECLSHADE-SPACMA124 on CEC2014 (Dim=50) and by ECLSHADE-SPACMA234 on CEC2017 (Dim=10). All algorithmic variants significantly surpass the original LSHADE-SPACMA, thereby validating the effectiveness of the individual strategies as well as their synergistic integration.

In terms of average performance, ECLSHADE-SPACMA2 exhibits the best performance among single-strategy variants, whereas ECLSHADE-SPACMA134 demonstrates the weakest performance within the multi-strategy combinations, suggesting that the exponential-function-based nonlinear population reduction mechanism plays the most critical role in enhancing overall optimization capability. Collectively, these findings consistently confirm the efficacy and complementary value of the proposed strategies and underscore the potential and practical applicability of the ECLSHADE-SPACMA framework.

4.7. Analysis of convergence performance and robustness

This section presents an evaluation of the convergence behavior and robustness of the ECLSHADE-SPACMA algorithm on the CEC14 and CEC17 benchmark test suites. To ensure clarity and conciseness, results are illustrated using convergence curves and box plots for six representative functions in the 100-dimensional case—specifically, the unimodal function F1, the simple multimodal function F9, the hybrid functions F15 and F18, and the composite functions F22 and F28.

Complete convergence profiles and box plots across all dimensions for both test suites are included in supplementary material, as shown in Figs. A.1 to A.16.

Figs. 9 and 10 present the convergence curves of the ECLSHADE-SPACMA algorithm in comparison with 13 other state-of-the-art algorithms on the CEC14 and CEC17 test suites, respectively. Experimental results indicate that on the CEC14 test suite, ECLSHADE-SPACMA achieves the highest convergence efficiency on functions F1, F9, F15, and F22. On function F18, its performance is surpassed only by HSES and mLSHADe-SPACMA, while on function F28, it is outperformed exclusively by EA4eig. For the CEC17 test suite, the algorithm demonstrates superior convergence efficiency on functions F9 and F15 and ranks among the top-performing methods on functions F18, F22, and F28. However, its convergence efficiency on function F1 is relatively lower. Overall, ECLSHADE-SPACMA delivers consistently strong convergence performance across the majority of benchmark functions.

The box plots depicted in Figs. 11 and 12 present a comparative analysis of the experimental results achieved by the ECLSHADE-SPACMA algorithm and 13 state-of-the-art algorithms over 51 independent runs on the CEC14 and CEC17 benchmark suites. The results show that ECLSHADE-SPACMA consistently yields a more favorable performance distribution compared to most competing algorithms on both test suites, with very few outliers evident. This suggests that the proposed algorithm exhibits strong robustness and high stability across a range of testing scenarios.

4.8. Wilcoxon signed-rank test analysis

In this section, a statistical analysis of the experimental results from the CEC2014 and CEC2017 test suites was performed using the non-parametric Wilcoxon signed-rank test. The detailed results of the test are provided in Tables A.3 to A.10 in the supplementary material. In addition, Tables A.3 to A.10 also present a comprehensive compilation of results from 51 independent runs for all algorithms on every test function in the CEC14 and CEC17 benchmark suites across all dimensions. These results include the mean values (Mean), standard deviations (Std), p-values derived from the Wilcoxon signed-rank test, the Wilcoxon overall performance metric (WP), Holm-corrected WP values (WP(Holm)), and the Friedman average ranks.

At a significance level of $\alpha = 0.05$, the outcomes are summarized in Tables 4 and 5, which present a comparative performance assessment between the proposed ECLSHADE-SPACMA algorithm and thirteen state-of-the-art algorithms—comprising nine CEC competition winners and four advanced variants: LSHADE, EBOwithCMAR, HSES, AGSK, LSHADE-SPACMA, ELSHADE-SPACMA, EA4eig, APSM-iSO, AL-SHADE, IDE-EDA, IMODE, mLSHADe-RL, and mLSHADe-SPACMA. Each benchmark function was independently run 51 times to ensure statistical robustness. In these tables, the symbols “+”, “-”, and “=” indicate that ECLSHADE-SPACMA statistically outperforms, underperforms, or performs comparably to the respective competitor, respectively. A higher number of “+” results reflects superior performance of the proposed algorithm, while a larger count of “-” results indicates relatively weaker performance.

Additionally, the results of the Wilcoxon signed-rank test are visually presented in Figs. 13 and 14 to facilitate clearer interpretation.

4.8.1. Wilcoxon analysis on CEC2014

This section presents a statistical evaluation of the CEC2014 benchmark results using the non-parametric Wilcoxon signed-rank test. The following provides a detailed and comprehensive analysis:

- (1) For the 10D scenario, ECLSHADE-SPACMA outperforms (underperforms) LSHADE, EBOwithCMAR, LSHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-iSO, AL-SHADE, mLSHADe-RL, and mLSHADe-SPACMA with counts of 12 (8), 5 (11), 8 (3), 17 (6), 9 (11), 21 (3), 20 (6), 11

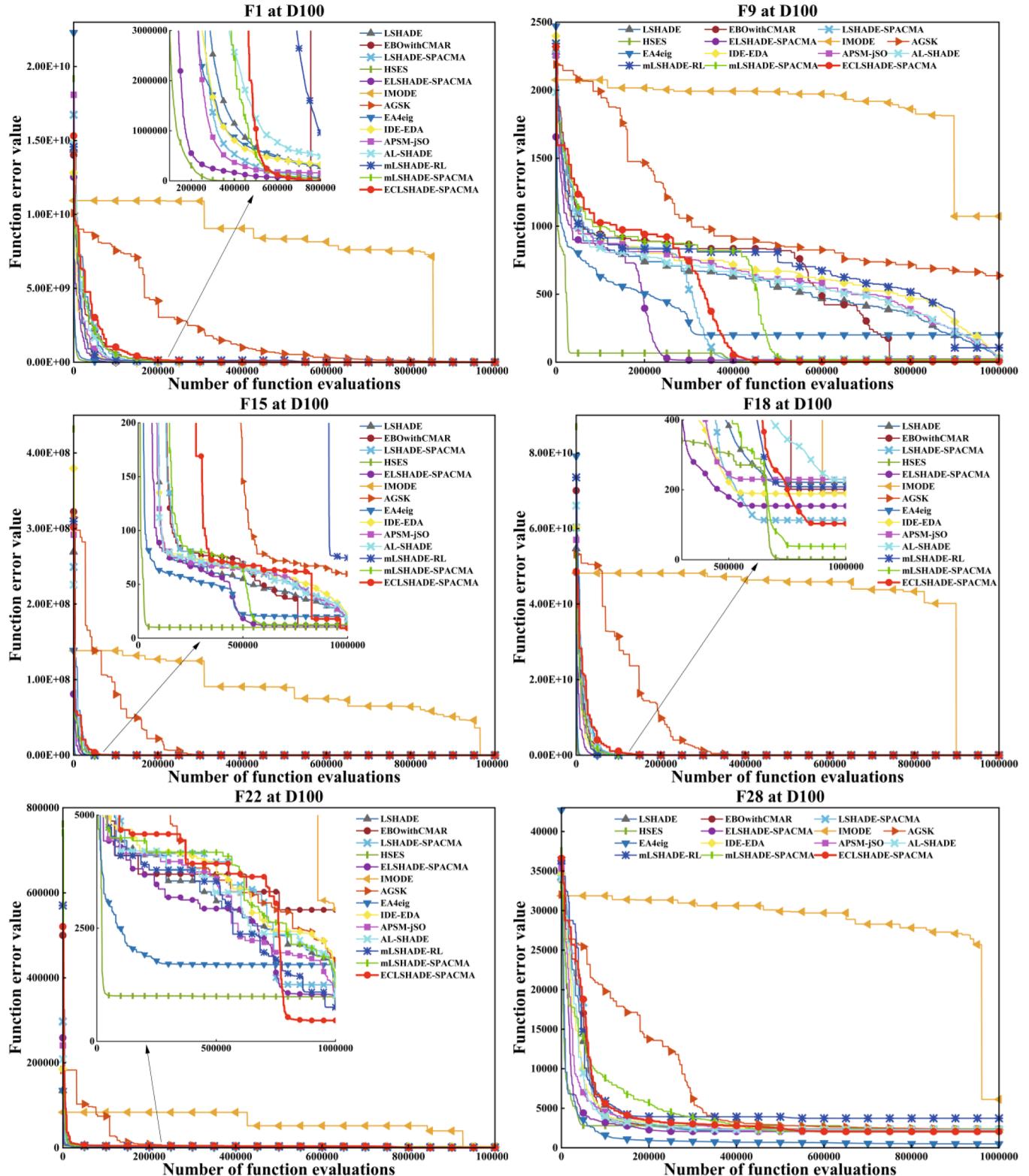


Fig. 9. The convergence curves of ECLSHADE-SPACMA are compared with state-of-the-art optimizers on the 100-D CEC2014.

- (11), 12 (7), 9 (6), 8 (9), 12 (4), and 11 (4), respectively. This indicates that ECLSHADE-SPACMA underperforms relative to EBOwithCMAR and ELSHADE-SPACMA, exhibits comparable performance to EA4eig, but surpasses the remaining ten algorithms in the 10D scenario.
(2) For the 30D scenario, ECLSHADE-SPACMA outperforms (underperforms) LSHADE, EBOwithCMAR, LSHADE-SPACMA, HSES,

ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-JSO, AL-SHADE, mLSHADE-RL, and mLSHADE-SPACMA with counts of 17 (2), 16 (3), 15 (3), 16 (8), 11 (6), 27 (2), 24 (1), 17 (7), 11 (9), 10 (6), 15 (3), 15 (6), and 13 (6), respectively. This demonstrates that ECLSHADE-SPACMA achieves superior performance compared to the other thirteen algorithms in the 30D scenario.

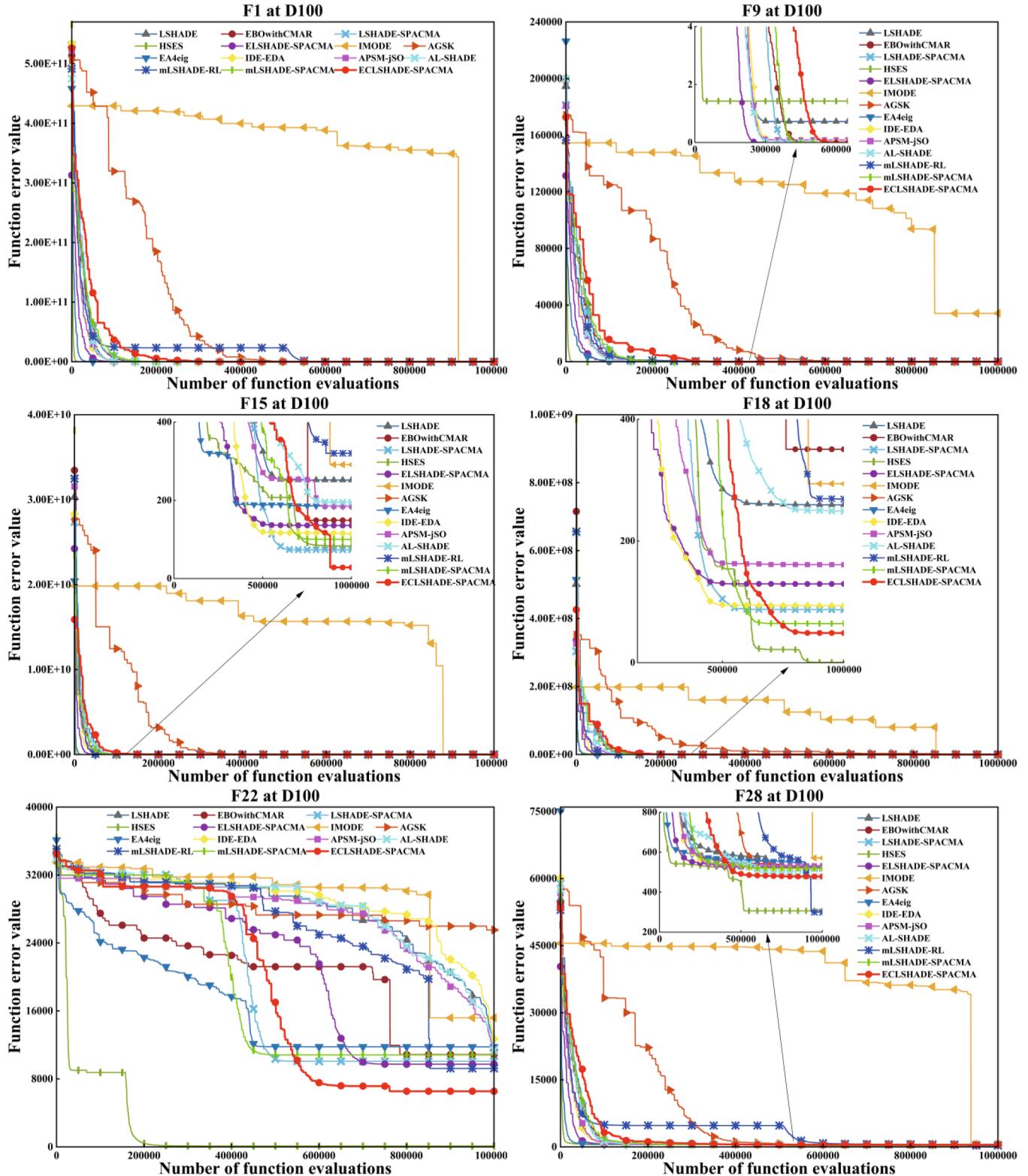


Fig. 10. The convergence curves of ECLSHADE-SPACMA are compared with state-of-the-art optimizers on the 100-D CEC2017.

- (3) For the 50D scenario, ECLSHADE-SPACMA outperforms (underperforms) LSHADE, EBOwithCMAR, LSHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mL SHADE-RL, and mL SHADE-SPACMA with counts of 18 (2), 15 (5), 13 (3), 15 (11), 15 (5), 28 (2), 25 (1), 17 (7), 17 (6), 16 (4), 18 (3), 17 (8), and 13 (9), respectively. This

confirms that ECLSHADE-SPACMA outperforms all thirteen algorithms in the 50D scenario.

- (4) For the 100D scenario, ECLSHADE-SPACMA outperforms (underperforms) LSHADE, EBOwithCMAR, LSHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mL SHADE-RL, and mL SHADE-SPACMA with counts of 19 (6), 19 (8), 13 (5), 11 (16), 12 (7), 27 (2), 27

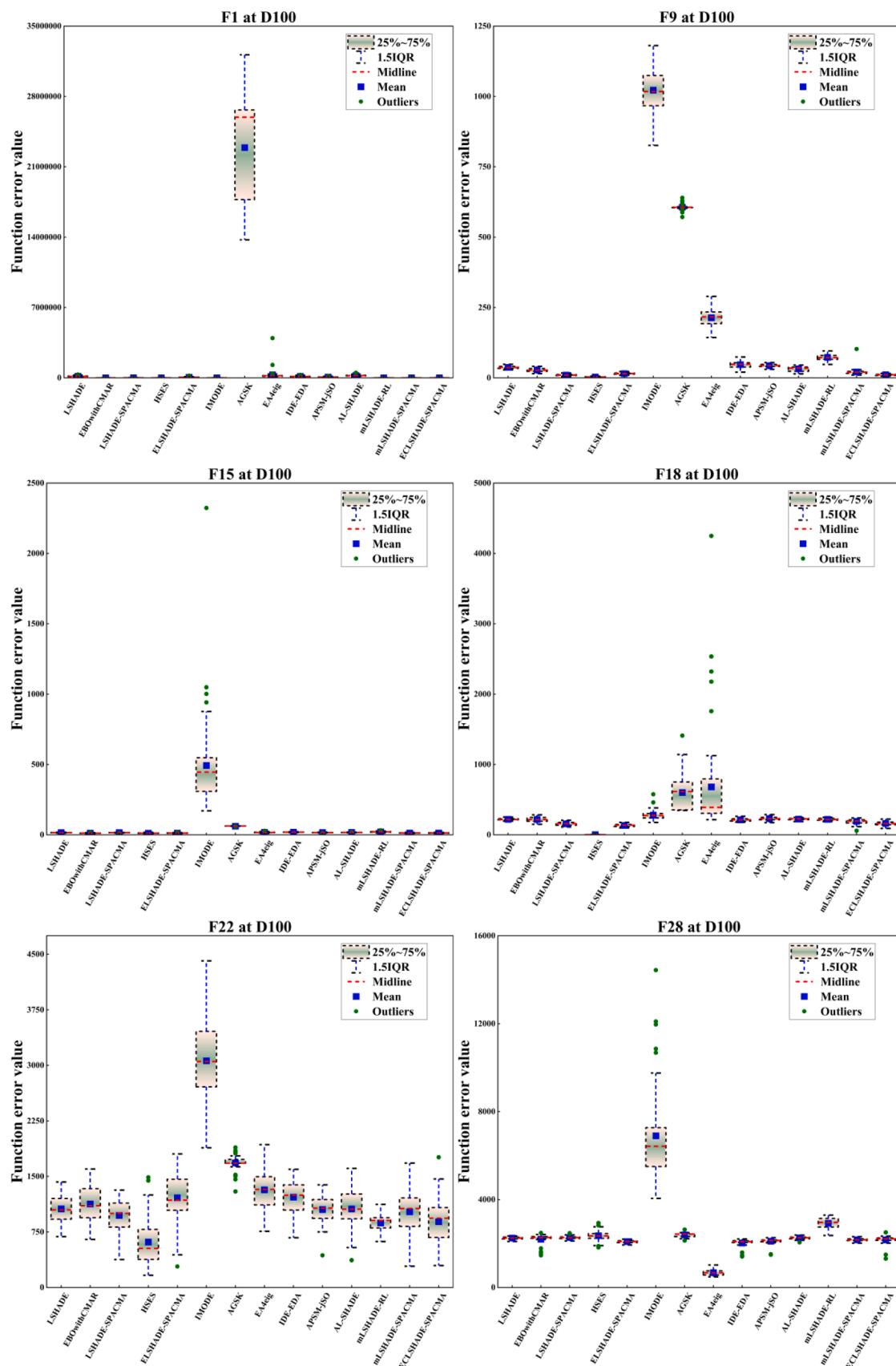


Fig. 11. The boxplot analysis of ECLSHADE-SPACMA is compared with state-of-the-art optimizers on the 100-D CEC2014.

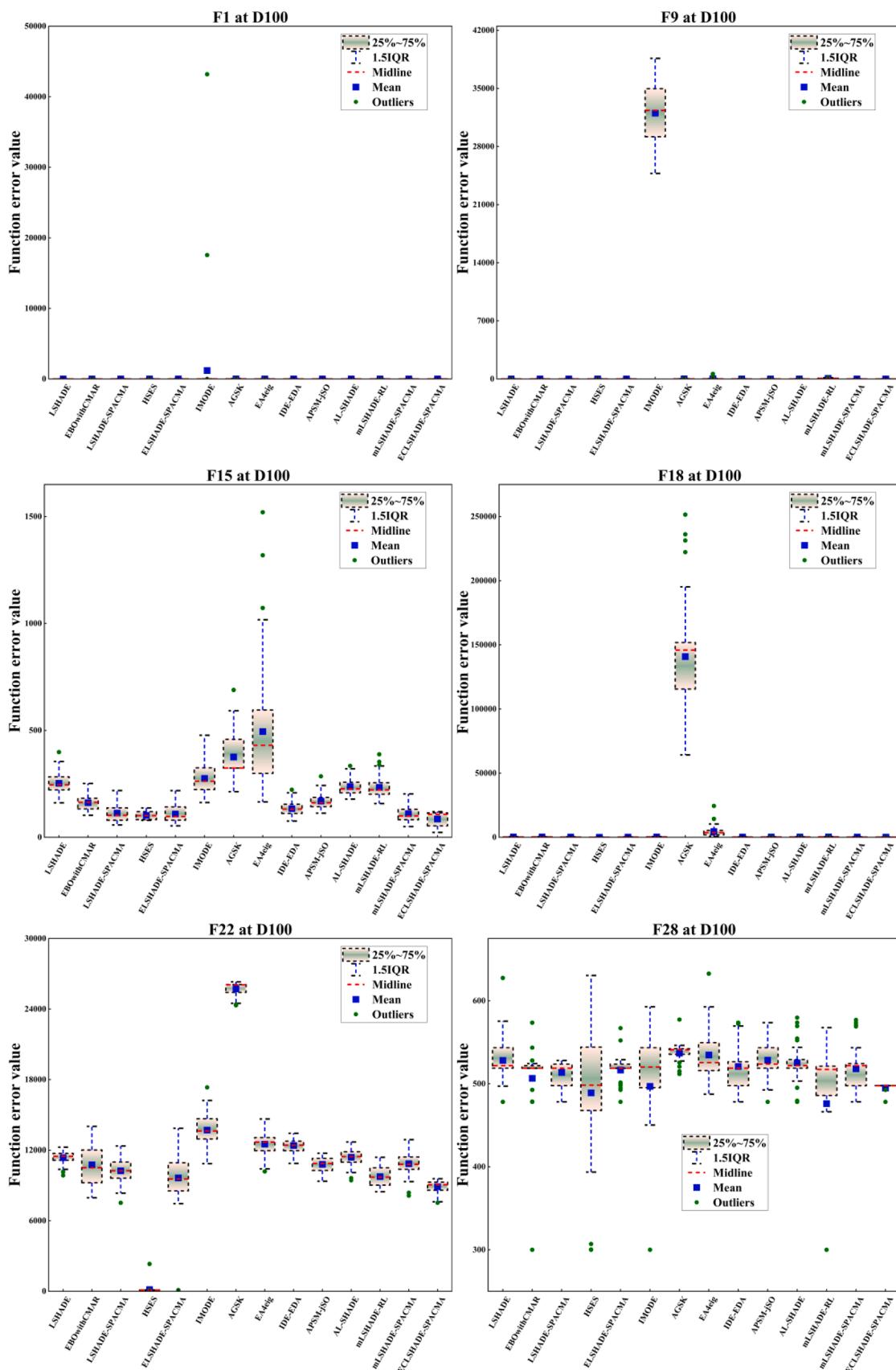


Fig. 12. The boxplot analysis of ECLSHADE-SPACMA is compared with state-of-the-art optimizers on the 100-D CEC2017.

Table 4

Wilcoxon signed-rank test outcomes for algorithm comparisons on the CEC2014 benchmark suite.

ECLSHADE-SPACMA vs.	Dim=10			Dim=30			Dim=50			Dim=100			Total 1		
	+	-	=	+	-	=	+	-	=	+	-	=	+	-	=
LSHADE	12	8	10	17	2	11	18	2	10	19	6	5	66	18	36
EBOwithCMAR	5	11	14	16	3	11	15	5	10	19	8	3	55	27	38
LSHADE-SPACMA	8	3	19	15	3	12	13	3	14	13	5	12	49	14	57
HSES	17	6	7	16	8	6	15	11	4	11	16	3	59	41	20
ELSHADE-SPACMA	9	11	10	11	6	13	15	5	10	12	7	11	47	29	44
IMODE	21	3	6	27	2	1	28	2	0	27	2	1	103	9	8
AGSK	20	6	4	24	1	5	25	1	4	27	2	1	96	10	14
EA4eig	11	11	8	17	7	6	17	7	6	18	10	2	63	35	22
IDE-EDA	12	7	11	11	9	10	17	6	7	16	6	8	56	28	36
APSM-jSO	9	6	15	10	6	14	16	4	10	16	7	7	51	23	46
AL-SHADE	8	9	13	15	3	12	18	3	9	20	6	4	61	21	38
mLSHADE-RL	12	4	14	15	6	9	17	8	5	18	9	3	62	27	31
mLSHADE-SPACMA	11	4	15	13	6	11	13	9	8	15	7	8	52	26	42
Total 2	155	89	146	207	62	121	227	66	97	231	91	68	820	308	432

Table 5

Wilcoxon signed-rank test results across algorithms on the CEC2017 Benchmark.

ECLSHADE-SPACMA vs.	Dim=10			Dim=30			Dim=50			Dim=100			Total 1		
	+	-	=	+	-	=	+	-	=	+	-	=	+	-	=
LSHADE	8	8	13	14	5	10	19	2	8	28	0	1	69	15	32
EBOwithCMAR	7	9	13	11	5	13	18	4	7	26	1	2	62	19	35
LSHADE-SPACMA	14	4	11	15	1	13	16	1	12	22	1	6	67	7	42
HSES	17	1	11	12	8	9	6	17	6	6	19	4	41	45	30
ELSHADE-SPACMA	10	3	16	14	5	10	14	2	13	23	2	4	61	12	43
IMODE	18	6	5	23	5	1	27	0	2	28	0	1	96	11	9
AGSK	14	6	9	26	0	3	29	0	0	29	0	0	98	6	12
EA4eig	7	11	11	14	10	5	20	2	7	26	2	1	67	25	24
IDE-EDA	10	4	15	9	5	15	16	3	10	25	1	3	60	13	43
APSM-jSO	10	6	13	10	4	15	14	2	13	26	1	2	60	13	43
AL-SHADE	9	10	10	16	5	8	20	3	6	28	0	1	73	18	25
mLSHADE-RL	13	6	10	15	4	10	20	5	4	23	2	4	71	17	28
mLSHADE-SPACMA	14	6	9	20	0	9	20	0	9	23	1	5	77	7	32
Total 2	151	80	146	199	57	121	239	41	97	313	30	34	902	208	398

(2), 18 (10), 16 (6), 16 (7), 20 (6), 18 (9), and 15 (7), respectively. This suggests that ECLSHADE-SPACMA underperforms relative to HSES but surpasses the remaining twelve algorithms in the 100D scenario.

From "Total 1" in Table 4, it is evident that ECLSHADE-SPACMA exhibits superior performance relative to the other 13 competing algorithms across a greater number of test functions. Specifically, ECLSHADE-SPACMA surpasses (is surpassed by) LSHADE, EBOwithCMAR, LSHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mLSHADE-RL, and mLSHADE-SPACMA on 66 (18), 55 (27), 49 (14), 59 (41), 47 (29), 103 (9), 96 (10), 63 (35), 56 (28), 51 (23), 61 (21), 62 (27), and 52 (26) test functions, respectively.

According to "Total 2", ECLSHADE-SPACMA shows enhanced performance (reduced performance) compared to the other 13 algorithms on 155 (89), 207 (62), 227 (66), and 231 (91) test functions in 10D, 30D, 50D, and 100D dimensions, respectively. The increase in victories correlates with the rise in dimensionality, underscoring its improved efficiency in addressing complex problems. This trend is visually verified in Fig. 13, which demonstrates the outstanding performance of ECLSHADE-SPACMA in higher dimensions within the CEC2014 benchmark test suite. Overall, the algorithm achieves 820 wins, incurs 308 losses, and results in 432 ties, highlighting its robust stability and effectiveness.

Furthermore, as shown in the analysis presented in Tables A.3 to A.6 (refer to the supplementary materials), for the majority of test functions across all four dimensions, the p-values obtained from the Wilcoxon signed-rank test were below the significance level α . Moreover, the

overall Wilcoxon performance metrics—namely WP and WP (Holm)—were also below α across all dimensional settings, with the single exception of the 10-dimensional case when compared against AL-SHADE, in which both metrics exceeded α . These findings demonstrate that, on the CEC14 benchmark suite, the performance of ECLSHADE-SPACMA is statistically significantly different from that of the other 13 competing algorithms.

4.8.2. Wilcoxon analysis on CEC2017

This section presents a statistical evaluation of the CEC2017 experimental outcomes using the non-parametric Wilcoxon signed-rank test. The following provides a detailed and comprehensive analysis:

- (1) For the 10D scenario, ECLSHADE-SPACMA outperforms (underperforms) LSHADE, EBOwithCMAR, LSHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mLSHADE-RL, and mLSHADE-SPACMA, with respective counts of 8 (8), 7 (9), 14 (4), 17 (1), 10 (3), 18 (6), 14 (6), 7 (11), 10 (4), 10 (6), 9 (10), 13 (6), and 14 (6). These results indicate that ECLSHADE-SPACMA performs less effectively than EBOwithCMAR, EA4eig, and AL-SHADE, shows comparable performance relative to LSHADE-SPACMA, and outperforms the remaining nine algorithms in the 10D scenario.
- (2) For the 30D scenario, ECLSHADE-SPACMA outperforms (underperforms) LSHADE, EBOwithCMAR, LSHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mLSHADE-RL, and mLSHADE-SPACMA with counts of 14 (5), 11 (5), 15 (1), 12 (8), 14 (5), 23 (5), 26 (0), 14 (10), 9 (5), 10 (4), 16 (5), 15 (4), and 20 (0), respectively. This

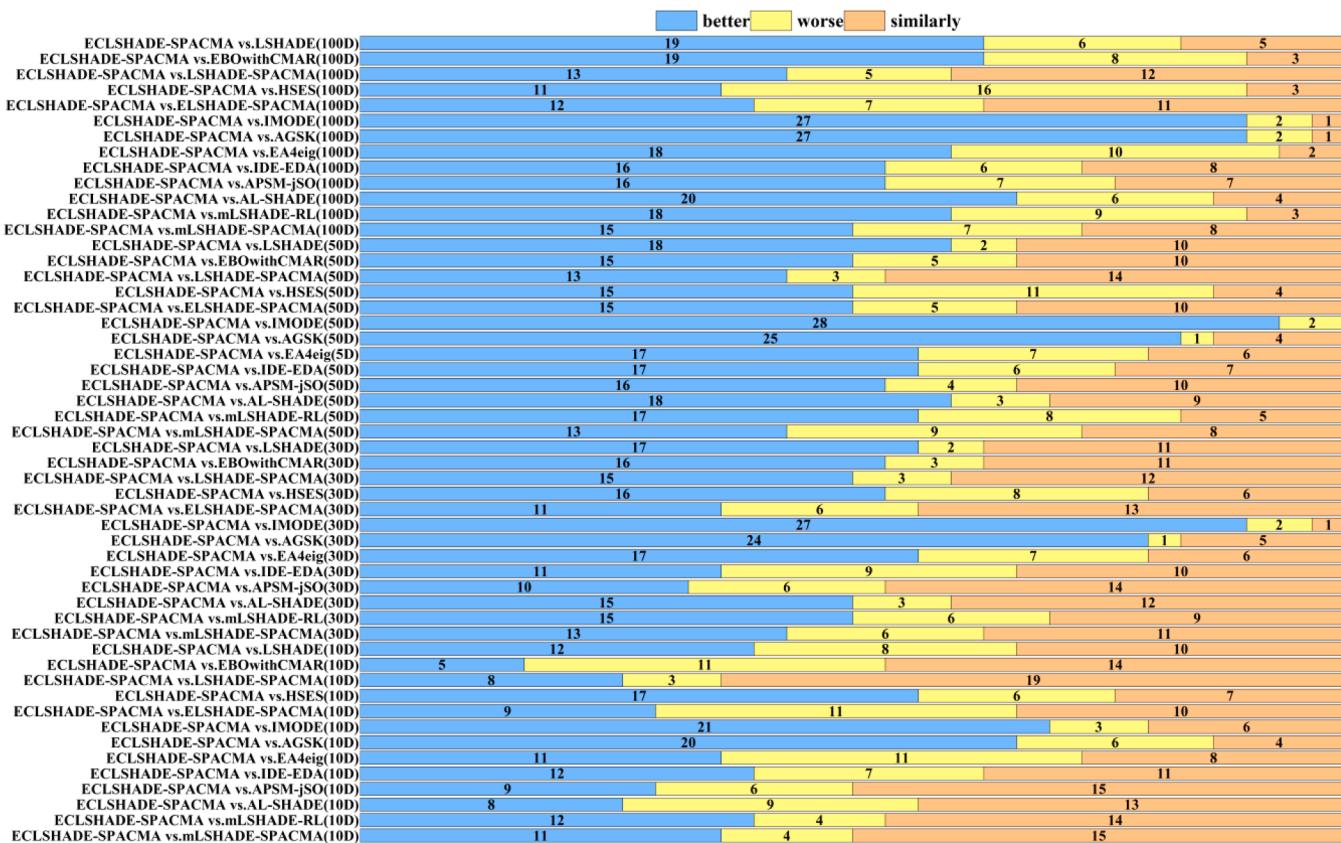


Fig. 13. Wilcoxon signed-rank test outcomes for the CEC2014 benchmark functions.

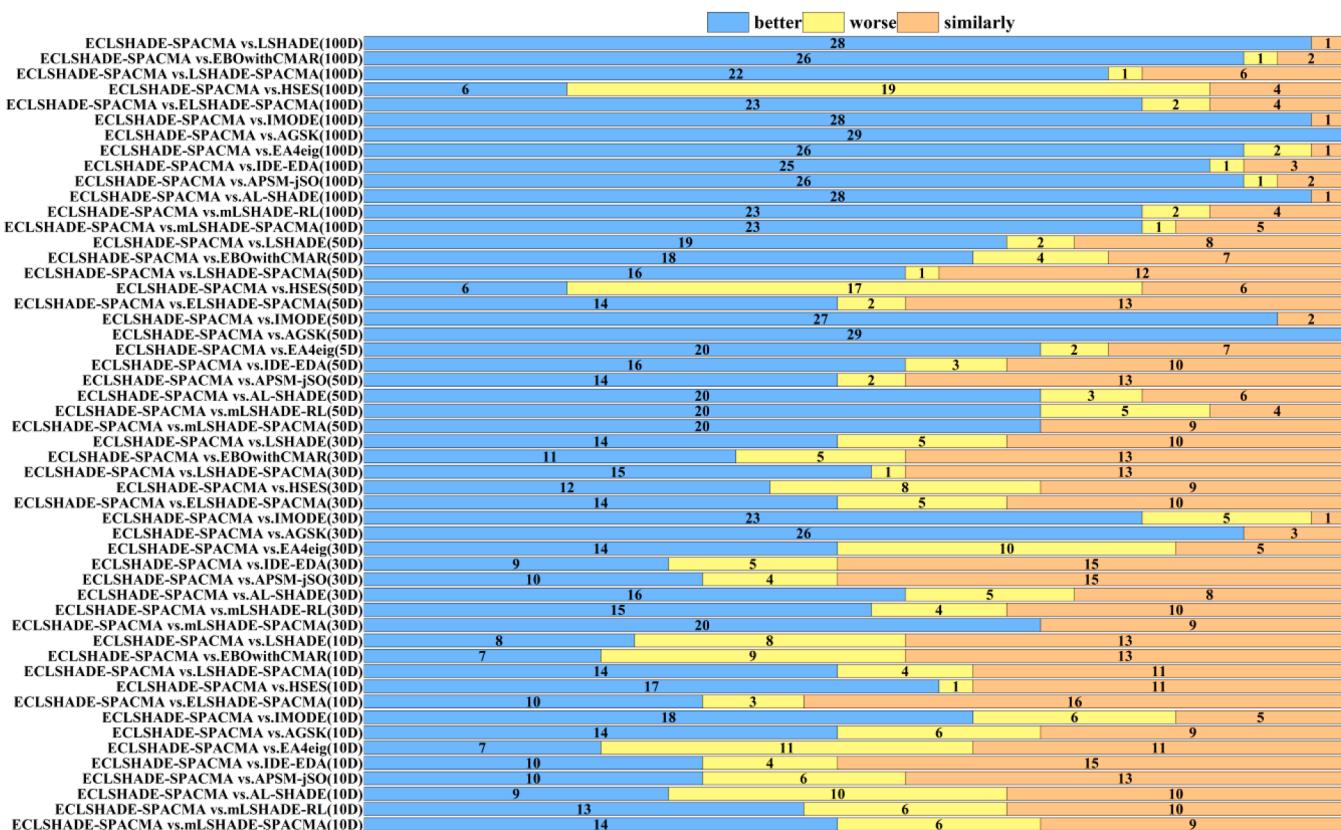


Fig. 14. Wilcoxon signed-rank test outcomes for the CEC2017 benchmark functions.

Table 6Friedman mean rank test results for CEC2017 and CEC2014 ($\alpha = 0.05$).

Algorithm	CEC2014				CEC2017				Mean	Mean Rank
	10D	30D	50D	100D	10D	30D	50D	100D		
LSHADE	7.6000	7.1667	7.6000	7.6500	8.0690	6.8793	7.5517	8.0517	7.5710	10
EBOwithCMAR	5.4333	6.7667	5.7500	6.7667	6.0000	5.8448	6.0172	6.7241	6.1629	2
LSHADE-SPACMA	7.4333	6.8167	5.9500	5.9000	8.3966	7.0517	6.6035	5.0000	6.6440	5
HSES	9.7333	8.1667	7.0333	6.1667	10.0345	7.1379	4.5000	3.3448	7.0147	8
ELSHADE-SPACMA	7.0333	6.0667	5.9167	5.1667	7.6379	6.9828	6.1897	5.6379	6.3290	3
IMODE	10.1667	12.5667	12.1833	11.7333	9.1035	12.4138	12.8276	12.5517	11.6933	14
AGSK	10.1000	11.1333	11.9000	11.9667	8.4483	11.9310	12.4828	12.8966	11.3573	13
EA4eig	5.9833	8.4667	7.6167	8.2333	4.8966	7.7931	9.3448	10.6897	7.8780	11
IDE-EDA	7.8500	6.6667	6.7833	7.6667	6.8966	5.3966	5.6035	6.8103	6.7092	6
APSM-jSO	6.9833	5.9333	6.8500	6.9000	7.0690	5.5862	5.6035	6.9483	6.4842	4
AL-SHADE	5.9500	6.5000	7.8500	8.3000	5.9310	6.4310	7.6207	9.2931	7.2345	9
mLSHADE-RL	7.5000	7.6333	7.9333	7.7333	8.8276	9.1552	9.4138	8.6897	8.3608	12
mLSHADE-SPACMA	7.5333	6.2000	6.7500	5.7500	8.5690	7.7586	7.3276	6.2241	7.0141	7
ECLSHADE-SPACMA	5.7000	4.9167	4.8833	5.0667	5.1207	4.6379	3.9138	2.1379	4.5471	1
Friedman-p-value	2.33E-08	3.47E-17	1.70E-16	2.11E-16	1.42E-08	3.91E-20	6.34E-28	1.29E-38	N/A	N/A

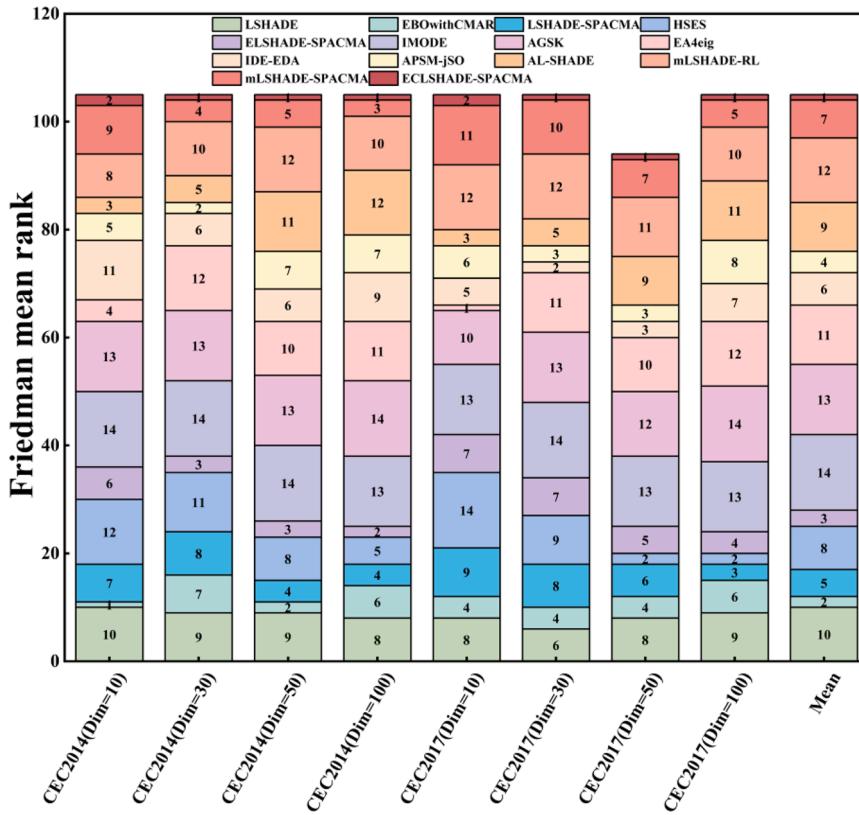


Fig. 15. Friedman mean rank comparisons across CEC2014 and CEC2017 benchmarks.

demonstrates that ECLSHADE-SPACMA achieves superior performance compared to the other thirteen algorithms in the 30D scenario.

- (3) For the 50D scenario, ECLSHADE-SPACMA outperforms (underperforms) LSHADE, EBOwithCMAR, LSHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mLSHADE-RL, and mLSHADE-SPACMA with counts of 19 (2), 18 (4), 16 (1), 6 (17), 14 (2), 27 (0), 29 (0), 20 (2), 16 (3), 14 (2), 20 (3), 20 (5), and 20 (0), respectively. This confirms that ECLSHADE-SPACMA underperforms relative to HSES but surpasses the remaining twelve algorithms in the 50D scenario.
- (4) For the 100D scenario, ECLSHADE-SPACMA outperforms (underperforms) LSHADE, EBOwithCMAR, LSHADE-SPACMA,

HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mLSHADE-RL, and mLSHADE-SPACMA with counts of 28 (0), 26 (1), 22 (1), 6 (19), 23 (2), 28 (0), 29 (0), 26 (2), 25 (1), 26 (1), 28 (0), 23 (2), and 23 (1), respectively. This suggests that ECLSHADE-SPACMA underperforms relative to HSES but surpasses the remaining twelve algorithms in the 100D scenario.

From “Total 1” in Table 5, it is observed that ECLSHADE-SPACMA achieves more wins compared to the other 12 competing algorithms, with performance slightly inferior only to HSES. Specifically, ECLSHADE-SPACMA outperforms (is outperformed by) LSHADE, EBOwithCMAR, LSHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mLSHADE-RL, and

Table 7

Computational complexity analysis of ECLSHADE-SPACMA and its competitive algorithms.

Algorithms	\hat{T}_2/s					$(\hat{T}_2 - T_1)/T_0$						
	Dim=10	Dim=30	Dim=50	Dim=100	Dim=10	Rank	Dim=30	Rank	Dim=50	Rank	Dim=100	Rank
LSHADE	0.5322	0.6063	0.7874	1.8826	11.0887	9	8.6223	7	9.2311	4	15.1550	4
EBOwithCMAR	0.8264	1.2284	1.9052	4.2201	17.8776	10	22.9760	10	35.0226	12	69.0900	13
LSHADE-SPACMA	1.3955	1.4948	1.5225	2.7834	31.0102	11	29.1230	11	26.1931	10	35.9391	8
HSES	0.3418	0.5778	1.1310	2.9580	6.6963	7	7.9643	6	17.1601	9	39.9681	11
ELSHADE-SPACMA	0.3143	0.4944	0.7007	1.7711	6.0607	5	6.0399	4	7.2296	3	12.5815	2
IMODE	0.2947	0.4443	0.8357	1.5827	5.6090	3	4.8835	3	10.3449	6	8.2333	1
AGSK	0.3163	0.6546	0.9758	2.9003	6.1072	6	9.7360	8	13.5779	7	38.6362	10
EA4eig	12.4516	23.5945	21.3287	16.5483	286.1201	14	539.0604	14	483.2065	14	353.5554	14
IDE-EDA	0.2999	0.5428	0.7885	1.9698	5.7297	4	7.1573	5	9.2566	5	17.1655	5
APSM-jSO	0.2311	0.4234	0.6884	1.8720	4.1404	2	4.4013	2	6.9453	2	14.9098	3
AL-SHADE	0.1777	0.4027	0.6857	1.9753	2.9089	1	3.9247	1	6.8841	1	17.2940	6
mLSHADE-RL	2.9437	1.8342	1.6810	2.8175	66.7337	13	36.9555	12	29.8506	11	36.7276	9
mLSHADE-SPACMA	0.3976	0.8177	1.0338	2.7529	7.9832	8	13.5007	9	14.9162	8	35.2362	7
ECLSHADE-SPACMA	1.8730	2.1418	2.2609	3.3267	42.0271	12	44.0524	13	43.2307	13	48.4766	12

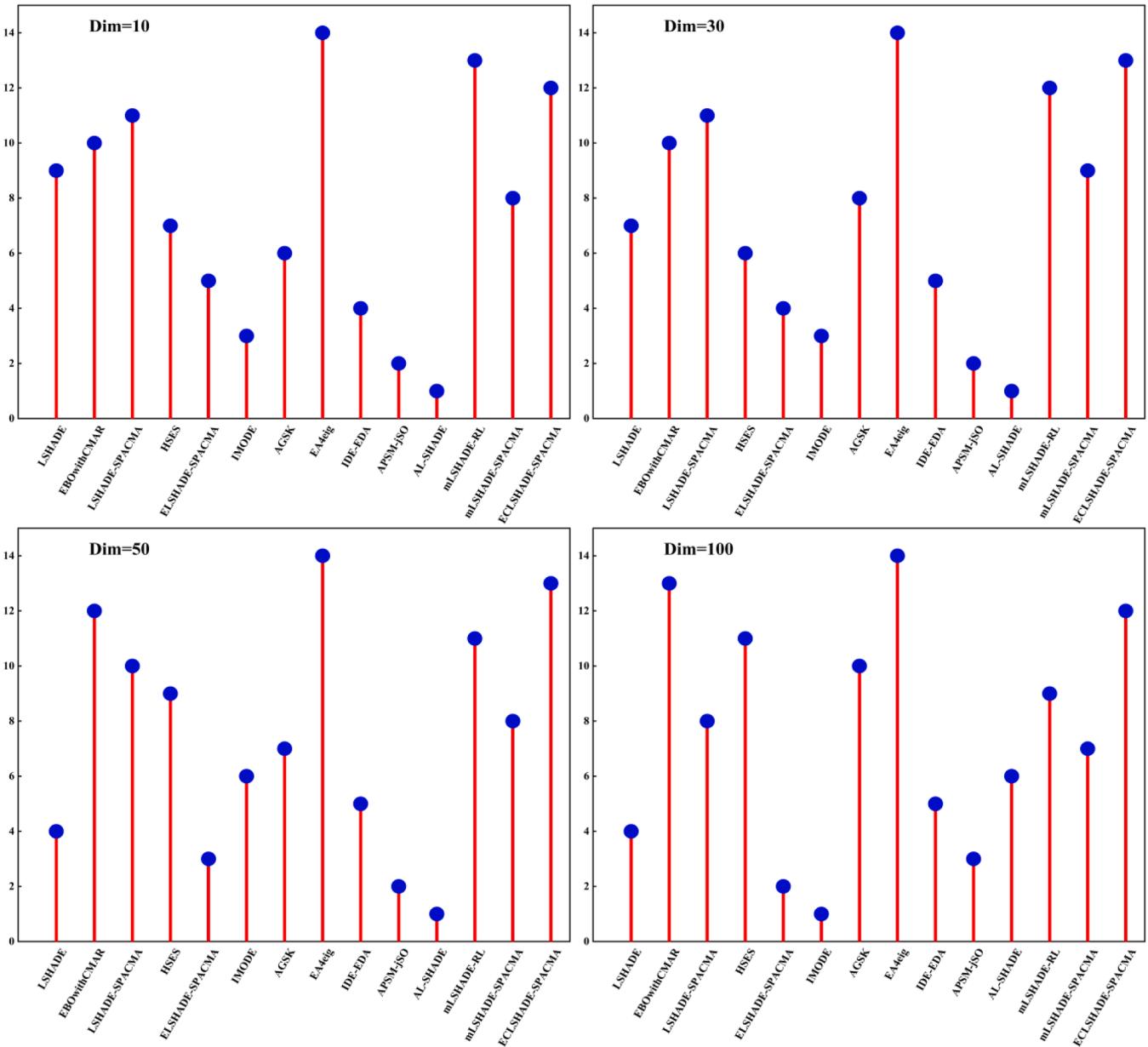
**Fig. 16.** Complexity rankings of ECLSHADE-SPACMA and competitive algorithms.

Table 8

Computational complexity analysis of ECLSHADE-SPACMA and its component-based variant.

Algorithms	\hat{T}_2/s				$(\hat{T}_2 - T_1)/T_0$							
	Dim=10	Dim=30	Dim=50	Dim=100	Dim=10	Rank	Dim=30	Rank	Dim=50	Rank	Dim=100	Rank
LSHADE-SPACMA	1.3955	1.4948	12.1581	1.9634	31.0102	2	29.1230	3	26.1931	3	35.9391	3
ECLSHADE-SPACMA1	1.4419	1.3682	12.9381	2.2700	32.0788	3	26.2022	2	19.1908	2	29.5544	2
ECLSHADE-SPACMA2	1.4979	1.5934	15.3587	3.0304	33.3724	4	31.3986	5	26.6581	4	43.7954	7
ECLSHADE-SPACMA3	1.7235	1.5772	9.5619	1.4336	38.5781	6	31.0247	4	26.7229	5	39.0153	5
ECLSHADE-SPACMA4	1.1768	1.1626	18.4106	4.1131	25.9621	1	21.4578	1	17.2506	1	28.8518	1
ECLSHADE-SPACMA123	1.9609	2.5284	2.6844	4.3642	44.0549	10	52.9745	10	53.0028	10	72.4157	10
ECLSHADE-SPACMA124	1.7777	1.6407	1.6126	2.8962	39.8283	8	32.4913	6	28.2725	6	38.5427	4
ECLSHADE-SPACMA134	1.5649	2.0774	1.7960	2.9588	34.9170	5	42.5660	8	32.5030	7	39.9865	6
ECLSHADE-SPACMA234	1.7268	1.7624	1.9920	3.4053	38.6527	7	35.2990	7	37.0252	8	50.2897	9
ECLSHADE-SPACMA	1.8730	2.1418	1.8730	2.0074	42.0271	9	44.0524	9	43.2307	9	48.4766	8

mLSHADE-SPACMA on 69 (15), 62 (19), 67 (7), 41 (45), 61 (12), 96 (11), 98 (6), 67 (25), 60 (13), 60 (13), 73 (18), 71 (17), and 77 (7) test functions, respectively.

According to the "Total 2" data, ECLSHADE-SPACMA exhibited superior (inferior) performance in 151 (80), 199 (57), 239 (41), and 313 (30) test cases, respectively, compared to the other 13 algorithms across the 10D, 30D, 50D, and 100D test functions. As dimensionality increases, the algorithm demonstrates a progressively enhanced advantage, with the number of successful outcomes increasing significantly while the number of unsuccessful outcomes decreasing correspondingly in high-dimensional scenarios, as visually corroborated by Fig. 14. This underscores its technical superiority. Furthermore, within the CEC2017 benchmark test suite comprising 1508 evaluations, the algorithm attained an impressive performance outcome of 902 wins, 208 losses, and 398 draws, thereby comprehensively illustrating its superior overall effectiveness.

Furthermore, a comparative analysis of the p-values, WP, and WP (Holm) values reported in Tables A.7 to A.10 and those in Tables A.3 to A.6 (see Supplementary Materials) indicates that the performance of ECLSHADE-SPACMA on the CEC17 benchmark suite demonstrates greater statistical significance in differences when compared to the 13 competing algorithms than on the CEC14 benchmark suite.

4.9. Friedman mean rank test analysis

This section utilizes the non-parametric Friedman mean rank test to assess the performance of various algorithms on the CEC2014 and CEC2017 benchmark function sets. The test results are summarized in Table 6, which provides performance rankings for 14 algorithms—LSHADE, EBOWithCMAR, LSHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mLSHADE-RL, mLSHADE-SPACMA, and ECLSHADE-SPACMA—based on 51 independent runs. As indicated in the final two columns of Table 6, the proposed ECLSHADE-SPACMA algorithm attains the lowest mean rank across all tested dimensions, demonstrating its superior overall performance. The EBOWithCMAR and ELSHADE-SPACMA algorithms follow in terms of performance ranking. Additionally, Fig. 15 presents a visual comparison of the relative rankings of the proposed algorithm against the other evaluated algorithms. A detailed analysis is provided below:

- (1) In the 10-dimensional scenario, on the CEC2014 test set, ECLSHADE-SPACMA ranked second, demonstrating slightly inferior performance compared to EBOWithCMAR but outperforming the remaining 12 algorithms. On the CEC2017 test set, ECLSHADE-SPACMA also ranked second, showing slightly lower performance than EA4eig but surpassing the other 12 algorithms.
- (2) For both the CEC2014 and CEC2017 test suites across 30-, 50-, and 100-dimensional problem instances, ECLSHADE-SPACMA consistently achieved the highest rank, demonstrating superior performance compared to the other 13 competing algorithms.

This highlights the algorithm's notable effectiveness in addressing complex optimization problems.

- (3) As indicated in the final row of Table 6, the Friedman test p-values for all dimensions of both the CEC2014 and CEC2017 benchmark suites are below the predetermined significance level ($\alpha = 0.05$). These statistically significant results justify the rejection of the null hypothesis, thereby confirming the presence of substantial performance differences between ECLSHADE-SPACMA and the other 13 competing algorithms across both test sets.
- (4) An integrated analysis of Table 6 and Fig. 15 indicates that the performance advantage of ECLSHADE-SPACMA becomes increasingly evident with increasing problem dimensionality. Overall, the proposed algorithm demonstrates a significant performance advantage over 13 competing algorithms on the CEC2014 and CEC2017 benchmark suites.

4.10. Computational complexity analysis

This paper conducts a computational complexity analysis of the proposed ECLSHADE-SPACMA algorithm based on the CEC2017 benchmark test suite [37]. The algorithm complexity is quantified by the formula $((\hat{T}_2 - T_1)/T_0)$, where the specific meanings of the parameters \hat{T}_2 , T_1 , and T_0 can be found in reference [37]. On a unified computing platform, the measured value of the base computing unit T_0 is 0.0433. The corresponding measured values of T_1 at dimensions of 10, 30, 50, and 100 are 0.0516, 0.2326, 0.3874, and 1.2258, respectively.

4.10.1. A computational complexity analysis of ECLSHADE-SPACMA and comparative algorithms

This section presents a comparative analysis of the computational complexity of ECLSHADE-SPACMA in comparison with 13 competing algorithms. Table 7 summarizes the evaluation results across dimensions of 10, 30, 50, and 100, while Fig. 16 provides a visual representation of the relative performance rankings. The computational complexity rankings of ECLSHADE-SPACMA are 12th, 13th, 13th, and 12th in these respective dimensions, indicating a relatively high computational cost. Nevertheless, the algorithm demonstrates significant advantages in optimization accuracy and convergence efficiency on the CEC2014 and CEC2017 benchmark test suites. Therefore, the additional computational overhead is justified and acceptable given its superior performance.

4.10.2. Computational complexity analysis of the core components in ECLSHADE-SPACMA

This section presents a systematic analysis of the computational complexity associated with each core component of ECLSHADE-SPACMA, aiming to identify key computational bottlenecks and to inform algorithmic configuration in resource-constrained environments. Building upon the eight ECLSHADE-SPACMA variants introduced in the ablation study detailed in Section 4.5, we assess the computational

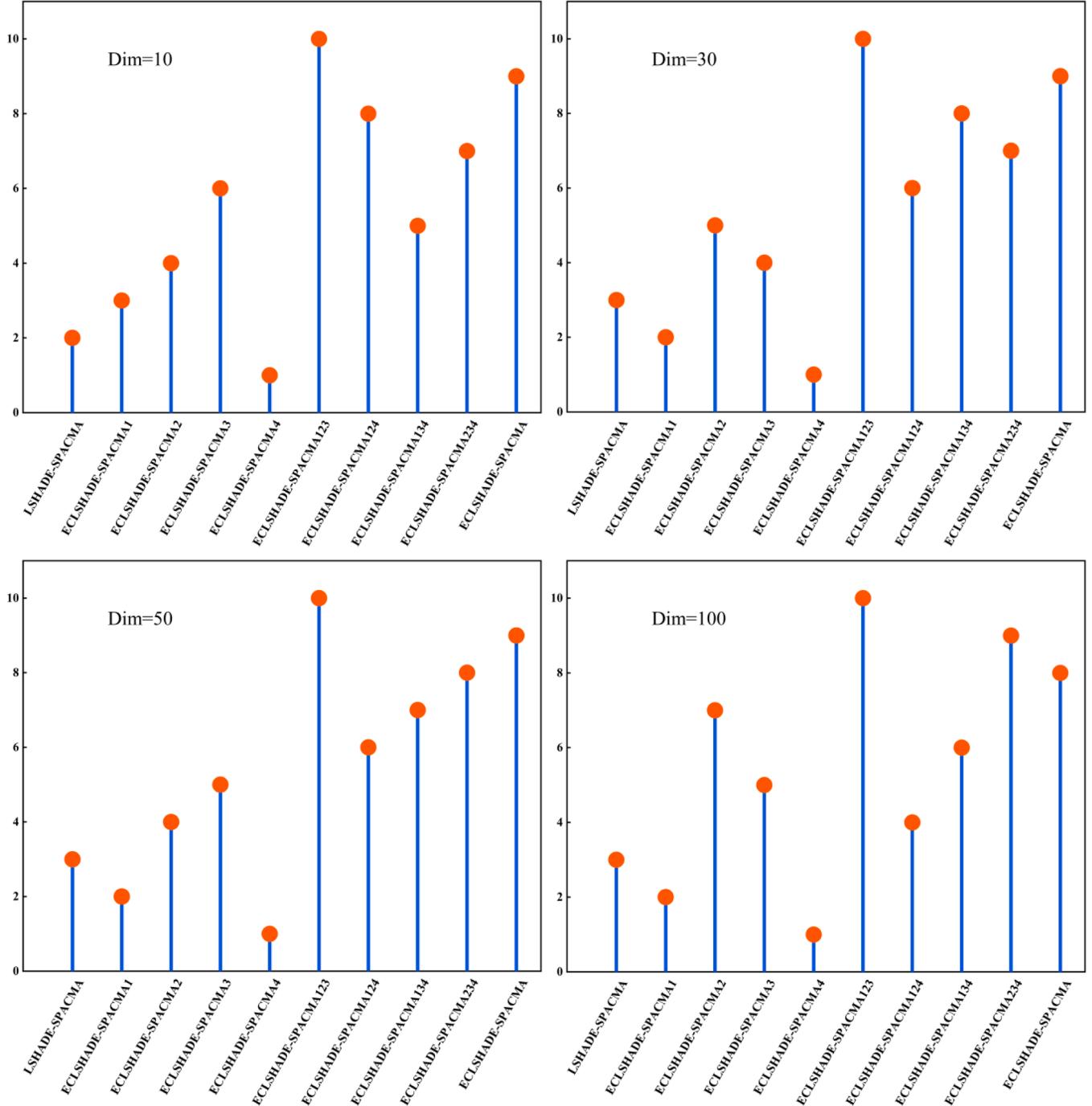


Fig. 17. Computational complexity comparison of ECLSHADE-SPACMA and its component-based variants.

overhead of individual components across multiple dimensional settings. The evaluation results for dimensions 10, 30, 50, and 100 are summarized in Table 8, while Fig. 17 offers a visual representation of the relative performance rankings among the variants.

The experimental results demonstrate that ECLSHADE-SPACMA4 consistently achieves the highest performance across all dimensions, whereas ECLSHADE-SPACMA123 exhibits the lowest performance under the same conditions. This contrast indicates that the adaptive probabilistic local search mechanism, guided by success rate feedback, effectively reduces the algorithm's overall computational cost. Among the single-strategy variants, ECLSHADE-SPACMA2 and ECLSHADE-SPACMA3 show comparatively weaker performance, while the multi-strategy configurations ECLSHADE-SPACMA134 and ECLSHADE-

SPACMA124 achieve notably superior outcomes. These findings suggest that the nonlinear population reduction strategy based on an exponential function and the elastic geometric weighted archive pruning strategy significantly increase computational overhead, thereby representing the primary contributors to the main computational bottleneck in ECLSHADE-SPACMA.

Furthermore, in resource-constrained applications, the primary design principle emphasizes optimization quality as the foremost objective, with computational cost as a secondary consideration. Comprehensive experimental evaluations on the CEC2014 and CEC2017 benchmark suites indicate that ECLSHADE-SPACMA outperforms 13 other state-of-the-art algorithms in both optimization accuracy and convergence efficiency, thereby validating the performance gains

Table 9
Obstacle details.

Obstacle	1	2	3	4	5	6	7
Coordinates	(15,65)	(40,25)	(50,50)	(50,80)	(85,50)	(110,20)	(125,60)
Radius	5	8	14	8	15	10	10

Table 10
Experimental assessment of UAV trajectory planning in complex mountainous terrain environments.

Algorithm	Best	Rank	Median	Rank	Worst	Rank	Mean	Rank	Std	Rank	Wilcoxon signed-rank test			
											+	-	=	P-value
LSHADE	77.6868	12	78.0502	8	78.1551	5	78.0341	8	0.1208	4	30	0	0	1.7344E-06
EBOwithCMAR	77.0176	3	78.1004	10	78.2443	8	77.9852	3	0.3144	6	27	3	0	1.7344E-06
LSHADE-SPACMA	77.6081	6	78.0252	3	87.0799	10	78.3766	10	1.7304	10	30	0	0	1.7344E-06
HSES	77.1613	4	78.1669	11	105.4680	14	80.2035	11	5.6483	14	30	0	0	1.9209E-06
ELSHADE-SPACMA	77.1657	5	78.0459	7	78.2043	7	78.0021	4	0.1903	5	29	1	0	1.7344E-06
IMODE	77.9405	13	78.7856	13	89.7835	12	80.5801	12	3.6359	11	30	0	0	8.3071E-04
AGSK	82.7572	14	90.8382	14	97.6484	13	91.0335	14	3.9166	12	30	0	0	1.7344E-06
EA4eig	76.9836	2	78.0182	2	78.1568	6	77.6612	2	0.4978	8	22	8	0	1.7344E-06
IDE-EDA	77.6254	8	78.0252	4	78.1252	3	78.0048	5	0.1026	2	30	0	0	7.6909E-06
APSM-JSO	77.6102	7	78.0278	5	78.1324	4	78.0083	6	0.1103	3	29	1	0	1.7344E-06
AL-SHADE	77.6853	11	78.0347	6	78.0869	2	78.0107	7	0.0994	1	30	0	0	1.7344E-06
mLSHADE-RL	77.6572	10	78.1790	12	81.1135	9	78.2457	9	0.5641	9	30	0	0	3.7243E-05
mLSHADE-SPACMA	77.6498	9	78.0547	9	87.1896	11	82.0535	13	4.4948	13	30	0	0	1.7344E-06
ECLSHADE-SPACMA	75.5439	1	77.6091	1	77.6317	1	77.4277	1	0.4284	7	NA	NA	NA	NA

achieved through the synergistic interaction of its four constituent components. Among these, the nonlinear population reduction strategy based on an exponential decay function (Component 2) plays the most critical role in determining overall algorithmic performance. To reduce computational demands without compromising solution quality, a more aggressive decay rate is recommended, enabling effective compression of the average population size. The elastic geometric weighted archive pruning strategy (Component 3) is indispensable for maintaining population diversity; its associated overhead can be effectively managed by reducing archive capacity and decreasing update frequency. In contrast, the dynamic fitness-directed selection pressure mutation strategy (Component 1) and the success-rate-driven adaptive probabilistic local search mechanism (Component 4) impose minimal computational burden while delivering consistent performance improvements, making them well-suited for integration as default mechanisms to support directional exploration and late-stage refinement.

Overall, to achieve an optimal trade-off between performance and efficiency under limited computational budgets, it is recommended to retain all four components with carefully calibrated intensities: Components (2) and (3) should operate at their minimal feasible levels to ensure solution quality, while Components (1) and (4) serve as efficient, low-cost supports for guidance and fine-tuning.

5. A simulation analysis of UAV path planning performance in complex mountainous terrain

5.1. Mathematical model establishment

Over the past decade, unmanned aerial vehicle (UAV) technology has found extensive applications across a broad spectrum of domains, including terrestrial communication [48], environmental surveillance [49–51], data acquisition [52], precision agriculture [53], emergency response operations [54,55], and military missions [56,57]. In scenarios characterized by complex terrain conditions, there exists a critical requirement for dependable three-dimensional path planning methodologies to guarantee both navigational efficacy and system resilience in UAV operations. Against this backdrop, this section presents an in-depth investigation into the application of the ECLSHADE-SPACMA algorithm for three-dimensional path planning in UAV systems. The mathematical model is detailed as follows:

A path planning framework is expressed based on the initial point $S(x_s, y_s, z_s)$, terminal point $T(x_t, y_t, z_t)$, and a set of n three-dimensional control points $\{P_1, P_2, \dots, P_n\}$. This framework employs a four-stage collaborative optimization procedure. In the first stage, discrete path points—including the starting point, control points, and terminal point—are interpolated into N continuous path points (x_i, y_i, z_i) using cubic spline interpolation. Following this transformation, the total path length objective function is mathematically expressed as:

$$F_{\text{path}} = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \quad (49)$$

This function quantifies the economic efficiency of the flight trajectory by summing the Euclidean distances between consecutive path points, thereby directly affecting energy consumption and mission completion time. In the second optimization stage, a flight altitude objective function is developed based on the identical set of path points. This function evaluates vertical fluctuations along the trajectory segments between the initial point and the terminal point by computing the standard deviation of altitude variations:

$$F_{\text{height}} = \sqrt{\frac{1}{N-2} \sum_{i=2}^{N-1} (z_i - \bar{z})^2} \quad (50)$$

$$\bar{z} = \frac{1}{N-2} \sum_{i=2}^{N-1} z_i \quad (51)$$

At this stage, the stability of the UAV's attitude in complex airflow environments is evaluated. In the third optimization stage, the flight deflection angle objective function is determined based on the sequence of path points. This function quantifies heading variations by analyzing the vector sequences formed by three consecutive path points:

$$F_{\text{angle}} = \sum_{k=3}^{N-1} |\theta_k - \theta_{k-1}| \quad (52)$$

$$\mathbf{A}_k = (x_k - x_{k-1}, y_k - y_{k-1}, z_k - z_{k-1}) \quad (53)$$

$$\mathbf{B}_k = (x_{k+1} - x_k, y_{k+1} - y_k, z_{k+1} - z_k) \quad (54)$$

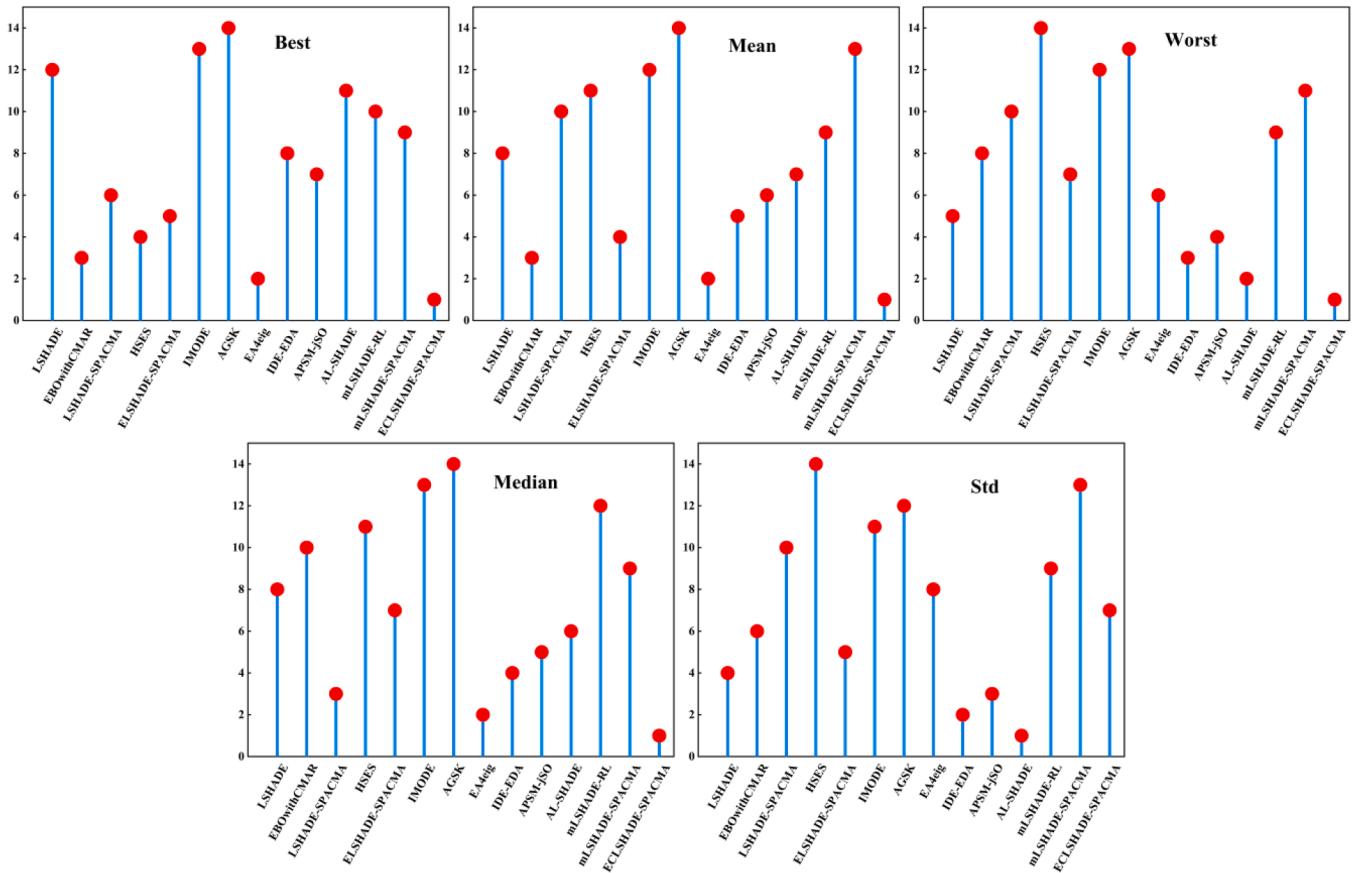


Fig. 18. Relative ranking comparison across evaluation metrics.

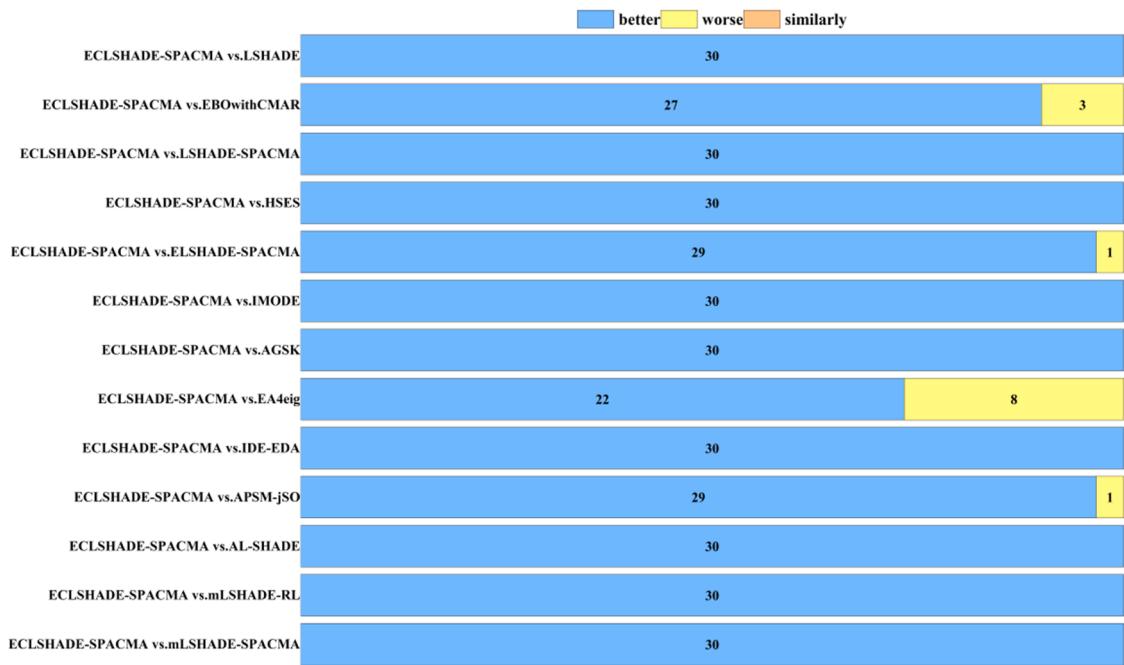


Fig. 19. Wilcoxon signed-rank test results for UAV trajectory planning in complex mountainous terrain environments.

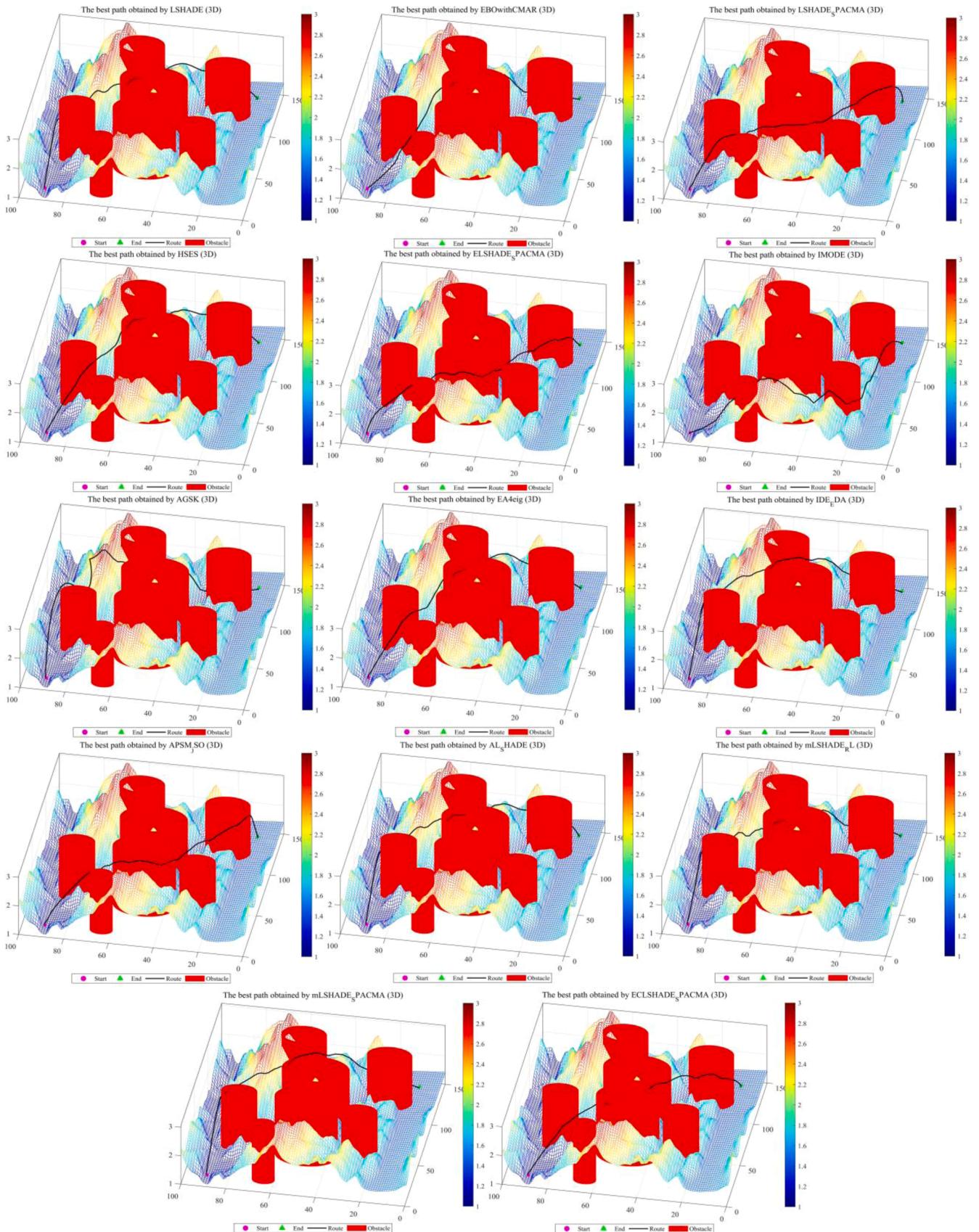


Fig. 20. UAV trajectory planning in complex mountainous terrain (three-dimensional visualization).

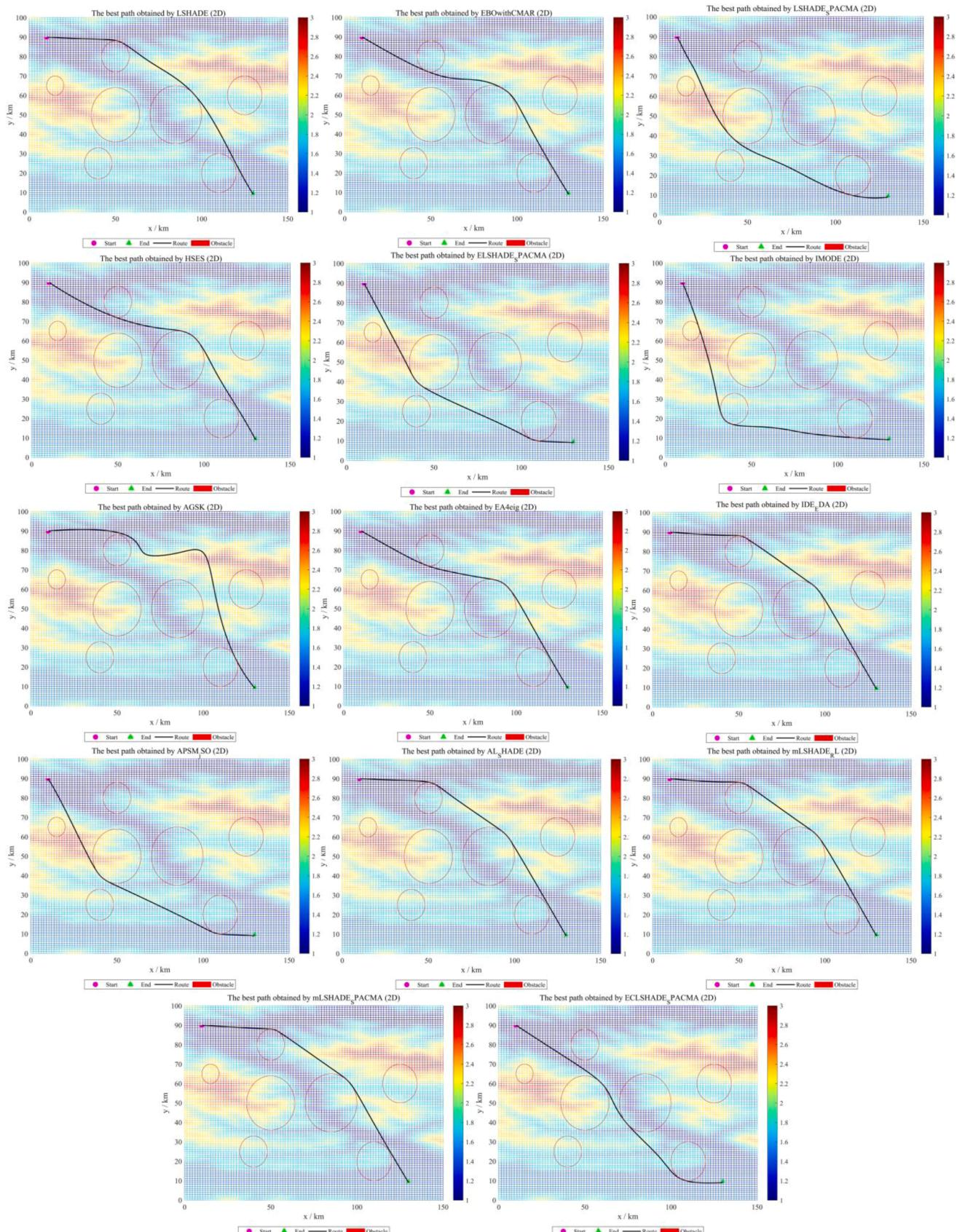


Fig. 21. UAV trajectory planning in complex mountainous terrain (two-dimensional-1).

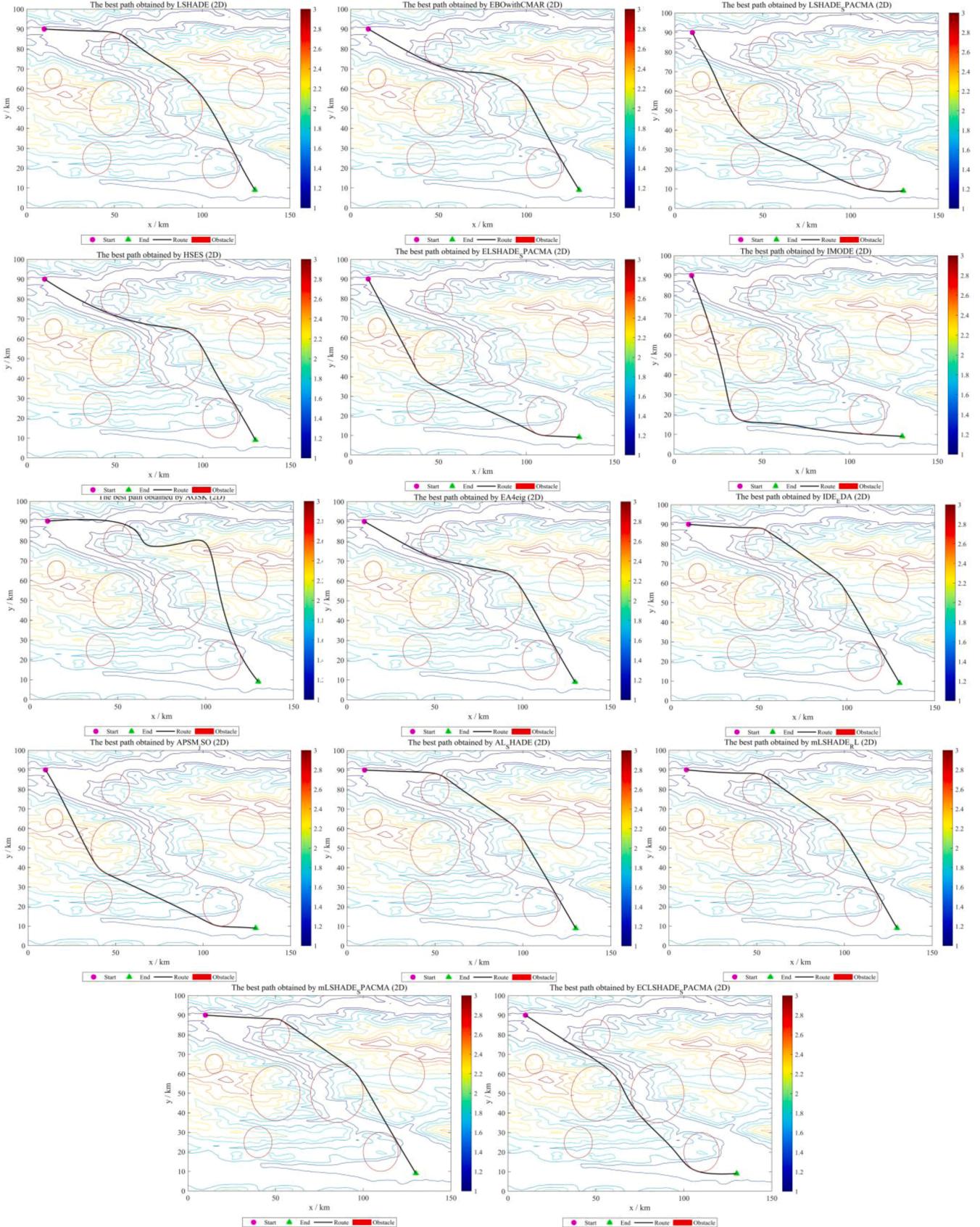


Fig. 22. UAV trajectory planning in complex mountainous terrain (two-dimensional-2).

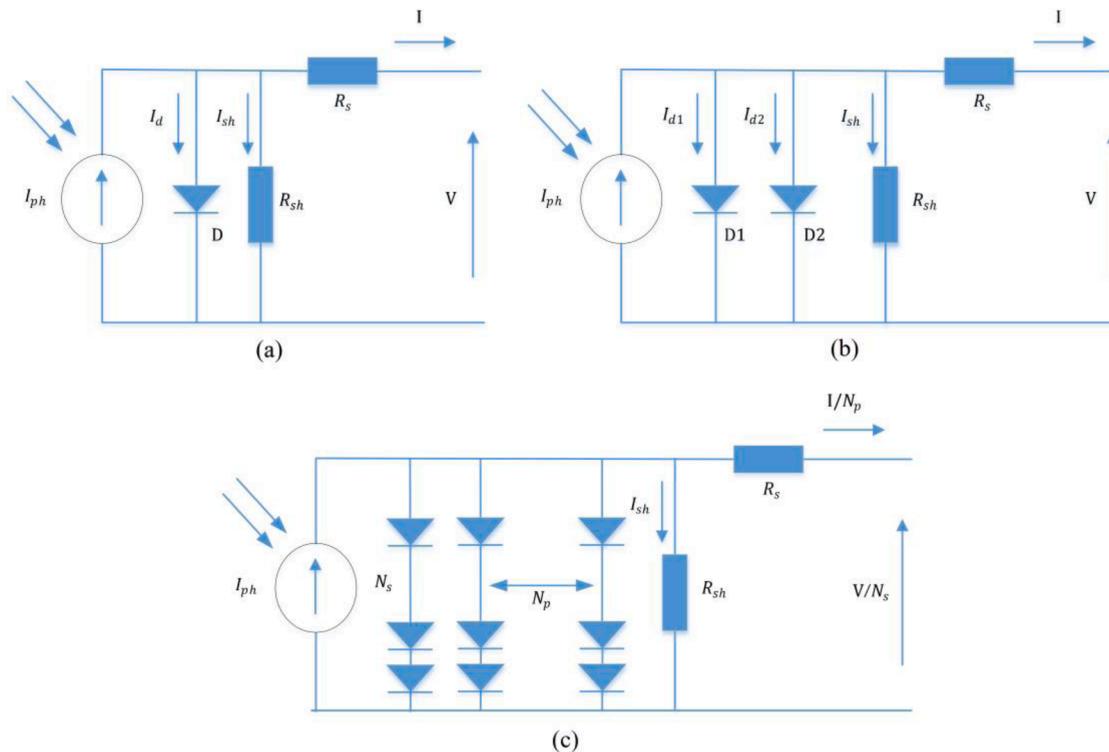


Fig. 23. Equivalent current diagrams for different types of photovoltaic models: (a) SDM, (b) DDM, (c) PV module model.

Table 11

The optimal RMSE values and the corresponding estimated parameter values for SDM.

Algorithm	$I_{ph}(A)$	$I_s(A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	a	RMSE
LSHADE	7.60775530343368E-01	3.23020813604709E-07	3.63770926498353E-02	5.37185238635120E+01	1.48118550081066E+00	9.86021877891504E-04
EBOwithCMAR	7.60775530359795E-01	3.23020799851200E-07	3.63770928727464E-02	5.37185240591565E+01	1.48118549646466E+00	9.86021877891492E-04
LSHADE-SPACMA	7.60775530431096E-01	3.23020799291006E-07	3.63770928527624E-02	5.37185224481762E+01	1.48118549633850E+00	9.86021877891478E-04
HSES	3.46320754078366E-01	8.11639064835216E-07	2.64931085177199E-01	9.28778989673024E+01	1.58108333059637E+00	1.54690014652660E-03
ELSHADE-SPACMA	7.60775530290203E-01	3.23020840212251E-07	3.63770923258722E-02	5.37185271419149E+01	1.48118550904683E+00	9.86021877891500E-04
IMODE	7.60775530150274E-01	3.23020845711481E-07	3.63770921514417E-02	5.37185245566293E+01	1.48118551092820E+00	9.86021877891553E-04
AGSK	7.60664107124098E-01	3.67521793164704E-07	3.58778041678650E-02	5.73423929965401E+01	1.49430551020801E+00	1.01923008917836E-03
EA4eig	7.60775530423222E-01	3.23020815448124E-07	3.63770925285312E-02	5.37185254000530E+01	1.48118550123845E+00	9.86021877891587E-04
IDE-EDA	7.60775530364867E-01	3.23020808368834E-07	3.63770926168183E-02	5.37185232733844E+01	1.48118549915739E+00	9.86021877891479E-04
APSM-jSO	7.60775530351505E-01	3.23020825869340E-07	3.63770924112050E-02	5.37185252563555E+01	1.48118550459077E+00	9.86021877891473E-04
AL-SHADE	7.60775530198557E-01	3.23020809549649E-07	3.63770926600277E-02	5.37185253850137E+01	1.48118549950203E+00	9.86021877891472E-04
mLSHADE-RL	7.60775530528379E-01	3.23020804087174E-07	3.63770928611197E-02	5.37185243888312E+01	1.48118549778403E+00	9.86021877891572E-04
mLSHADE-SPACMA	7.60775530253890E-01	3.23020811623083E-07	3.63770926558305E-02	5.37185254055154E+01	1.48118550012888E+00	9.86021877891470E-04
ECLSHADE-SPACMA	7.60775530440534E-01	3.23020816611780E-07	3.63770926651907E-02	5.37185242376954E+01	1.48118550172216E+00	9.86021877891469E-04

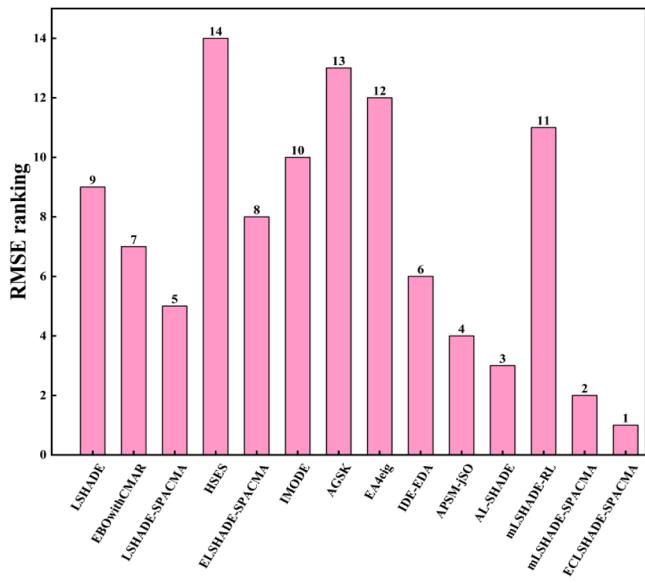


Fig. 24. RMSE-based performance evaluation and ranking of algorithms for SDM.

$$\theta_k = \arccos\left(\frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|}\right) \quad (55)$$

This computational process exerts a direct impact on both the mechanical wear of the aerial vehicle and the accuracy of trajectory tracking. Following this, the outputs from the first three stages are integrated into a formulated weighted total cost function:

$$F_{\text{total}} = \omega_1 F_{\text{path}} + \omega_2 F_{\text{height}} + \omega_3 F_{\text{angle}} \quad (56)$$

Here, $\omega_1 = 0.5$, $\omega_2 = 0.3$, $\omega_3 = 0.2$, and the sum of these weighting coefficients equals one.

5.2. Comparative analysis in relation to competing algorithms

This section evaluates the practical performance of 14 optimization algorithms—LSHADE, EBOwithCMAR, LSHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mLSHADE-RL, mLSHADE-SPACMA, and ECLSHADE-SPACMA—in solving UAV path planning problems within complex mountainous terrain. The flight origin is defined as (10, 90, 1.11), and the destination is set at (130, 9, 1.37). Obstacle configurations are based on the parameters outlined in Table 9.

To ensure fair and consistent evaluation, all comparative algorithms are configured with an identical maximum evaluation budget of 25,000 function evaluations per run and are independently executed 30 times to enhance statistical reliability. Five performance metrics—Best, Median, Worst, Mean, and Standard Deviation (Std)—are recorded to comprehensively evaluate the effectiveness and robustness of the algorithms. The results are systematically summarized, ranked according to performance, and uniformly presented in Table 10, where optimal values are indicated in bold. Fig. 18 provides a visual comparison of the relative rankings of each algorithm across the five performance indicators.

Additionally, to assess the statistical significance of performance differences, a wilcoxon signed-rank test is performed at a significance level of $\alpha = 0.05$ based on the 30 independent runs. The outcomes of this test are reported in Table 10 and further illustrated in Fig. 19.

The performance analysis reveals that, although ECLSHADE-SPACMA ranks sixth in the Std metric—indicating relatively lower stability—it outperforms all competing algorithms on the other four key performance indicators: Best, Median, Worst, and Mean.

Furthermore, the results of the Wilcoxon signed-rank test based on

30 independent runs show that ECLSHADE-SPACMA achieves superior outcomes compared to EBOwithCMAR, ELSHADE-SPACMA, EA4eig, and APSM-jSO in 27, 29, 22, and 29 runs, respectively, and consistently exceeds each of the remaining nine competing algorithms across all 30 runs.

Additionally, as indicated in the final column of Table 10, all corresponding p-values are below the predetermined significance level of $\alpha = 0.05$. Taken together, these results confirm that ECLSHADE-SPACMA not only demonstrates notable performance superiority in unmanned aerial vehicle path planning under complex mountainous terrain but also exhibits statistically significant advantages.

The three-dimensional and two-dimensional top-down views of the optimal flight paths generated by ECLSHADE-SPACMA and the comparative algorithms in complex mountainous terrain are presented in Figs. 20, 21, and 22, respectively. The proposed methodology achieves significant enhancements in trajectory quality. Based on comprehensive evaluations, ECLSHADE-SPACMA yields excellent numerical optimization outcomes while demonstrating strong practical applicability for engineering challenges such as unmanned aerial vehicle route planning. Thus, it can be regarded as a robust and versatile optimization framework.

6. Simulation analysis of parameter identification for solar photovoltaic models

With the accelerating global energy transition, solar photovoltaic (PV) technology has become an integral component of modern energy systems due to its clean and renewable characteristics [58], playing a crucial role in reducing carbon emissions and achieving energy independence [59,60]. To fully exploit the performance potential of PV systems, it is essential to establish accurate mathematical models. The core of this modeling effort lies in the precise estimation of unknown cell parameters, which critically affects model simulation accuracy, the optimization of system control strategies, the efficiency of power converter design, and the reliability of performance monitoring [61].

Constructing high-fidelity PV models serves as the foundation for subsequent research and practical applications. Currently, the single diode model [62] (SDM), double diode model [63] (DDM), and PV module model [64] represent the most widely adopted and extensively studied mainstream models. This study will establish corresponding mathematical modeling frameworks based on these three mainstream models, providing a theoretical foundation for subsequent parameter identification, performance analysis, and system optimization.

6.1. Mathematical modelling of PV cells

6.1.1. Single diode model

The single diode model (SDM), whose equivalent circuit is illustrated in Fig. 23(a), characterizes photovoltaic cells using five essential parameters: the light-generated current (I_{ph}), the diode reverse saturation current (I_s), the diode ideality factor (a), the series resistance (R_s), which accounts for resistive losses in series connections, and the shunt resistance (R_{sh}), which represents leakage currents in parallel branches. The output current I can be calculated by subtracting the diode current (I_d) and the shunt resistor current (I_{sh}) from I_{ph} :

$$I = I_{ph} - I_d - I_{sh} \quad (57)$$

The diode current I_d is governed by the following expression:

$$I_d = I_s \left[\exp\left(\frac{(V + IR_s) \cdot q}{akT}\right) - 1 \right] \quad (58)$$

where V denotes the output voltage, and the shunt current I_{sh} adheres to Ohm's law:

Table 12

The optimal RMSE values and the corresponding estimated parameter values for DDM.

Algorithm	$I_{ph}(A)$	$I_{s1}(A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	a_1	$I_{s2}(A)$	a_2	RMSE
LSHADE	7.60781079133518E-01	2.25974243656000E-07	3.67404286411299E-02	5.54854284403589E+01	1.45101864947594E+00	7.49340199313124E-07	1.99999999999977E+00	9.82484881636855E-04
EBOwithCMAR	7.60780530058645E-01	7.22407260124507E-07	3.67233323204134E-02	5.548562035210570E+01	1.99916286722829E+00	2.29044313592542E-07	1.45216344285851E+00	9.82500580948834E-04
LSHADE-SPACMA	7.60781078990261E-01	2.25974446665895E-07	3.67404277175777E-02	5.54854303381488E+01	1.45101872521646E+00	7.49338591486389E-07	1.99999999999967E+00	9.82484881636891E-04
HSES	6.76753260044732E-01	4.84472637651242E-07	4.64211821772257E-01	5.17241284492072E-01	1.07246963124372E+00	2.19935365077078E-07	1.22566823066476E+00	2.20975946140025E-03
ELSHADE-SPACMA	7.60781079050393E-01	2.25974123495033E-07	3.67404292280894E-02	5.54854317951080E+01	1.45101860474532E+00	7.49341183638063E-07	1.99999999999999E+00	9.82484881636910E-04
IMODE	7.60758113807450E-01	2.01180056977635E-07	3.67021828851940E-02	5.62593615448518E+01	1.44309314005158E+00	6.66009801445122E-07	1.89416587759642E+00	9.84806223741342E-04
AGSK	7.60467806308456E-01	9.68213487887121E-08	3.74798810792625E-02	6.01690644504646E+01	1.41042828655354E+00	1.99565388954587E-07	1.53229382912594E+00	1.54926374239434E-03
EA4eig	7.60768230218367E-01	2.85679486875983E-07	3.64886624898867E-02	5.47244202159590E+01	1.99284059276609E+00	2.84471561571294E-07	1.47049003710714E+00	9.84090413602830E-04
IDE-EDA	7.60781078913233E-01	2.25974161424275E-07	3.67404292756890E-02	5.54854342820375E+01	1.45101861858585E+00	7.49340838223398E-07	1.99999999999599E+00	9.82484881637015E-04
APSM-jSO	7.60781079296712E-01	2.25974216386442E-07	3.67404290630093E-02	5.54854303782598E+01	1.45101863891947E+00	7.49340374976937E-07	1.99999999999695E+00	9.82484881637017E-04
AL-SHADE	7.60781079091499E-01	2.25974180139803E-07	3.67404291590912E-02	5.54854319032408E+01	1.45101862570597E+00	7.49340709323576E-07	1.9999999999997E+00	9.82484881636879E-04
mLSHADE-RL	7.60778220502427E-01	2.56601917384262E-07	3.64718945491050E-02	5.40561646735959E+01	1.46529835048246E+00	1.32423323344547E-07	1.68635690301842E+00	9.83467722328568E-04
mLSHADE-SPACMA	7.60781079022176E-01	7.49339054755178E-07	3.67404280539535E-02	5.54854285486851E+01	1.99999999999944E+00	2.25974379913737E-07	1.45101869987655E+00	9.82484881636907E-04
ECLSHADE-SPACMA	7.60781079040845E-01	2.25973764139095E-07	3.67404315156506E-02	5.54854311750917E+01	1.45101846921812E+00	7.49343839700760E-07	1.99999999999864E+00	9.82484881636854E-04

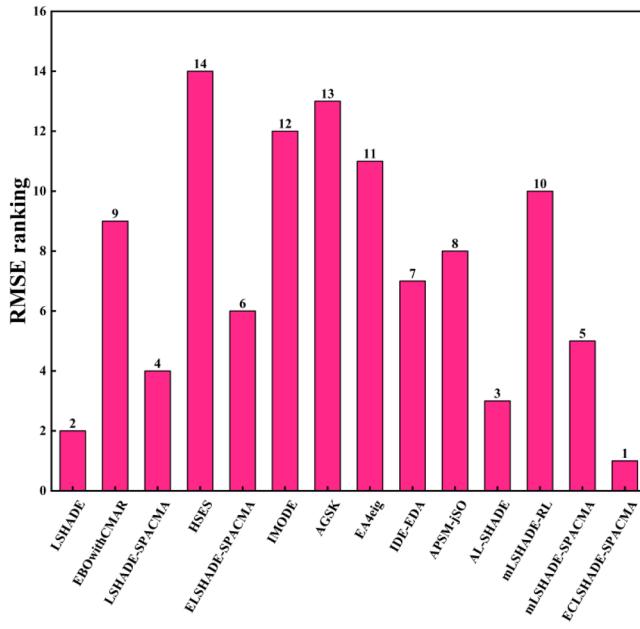


Fig. 25. RMSE-based performance evaluation and ranking of algorithms for DDM.

$$I_{sh} = \frac{V + IR_s}{R_{sh}} \quad (59)$$

Here, q denotes the elementary charge, k represents the Boltzmann constant, and T refers to the absolute temperature. By integrating these relationships, the implicit expression of the SDM is derived:

$$I = I_{ph} - I_s \left[\exp\left(\frac{(V + IR_s) \cdot q}{akT}\right) - 1 \right] - \frac{V + IR_s}{R_{sh}} \quad (60)$$

6.1.2. Double diode model

The more complex double diode model (DDM), with its equivalent circuit illustrated in Fig. 23(b), extends the SDM by incorporating an additional diode to account for recombination losses in the depletion region. This model requires seven parameters: the light-generated current I_{ph} , the reverse saturation currents of the two diodes I_{s1} and I_{s2} , their respective ideality factors a_1 and a_2 , the series resistance R_s , and the shunt resistance R_{sh} . The output current I is obtained by subtracting both diode currents I_{d1} and I_{d2} from I_{ph} :

$$I = I_{ph} - I_{d1} - I_{d2} - I_{sh} \quad (61)$$

The current through the first diode I_{d1} is mathematically expressed as:

$$I_{d1} = I_{s1} \left[\exp\left(\frac{(V + IR_s) \cdot q}{a_1 kT}\right) - 1 \right] \quad (62)$$

while the second diode current I_{d2} is expressed as:

$$I_{d2} = I_{s2} \left[\exp\left(\frac{(V + IR_s) \cdot q}{a_2 kT}\right) - 1 \right] \quad (63)$$

leading to the complete expression of the DDM:

$$I = I_{ph} - I_{s1} \left[\exp\left(\frac{(V + IR_s) \cdot q}{a_1 kT}\right) - 1 \right] - I_{s2} \left[\exp\left(\frac{(V + IR_s) \cdot q}{a_2 kT}\right) - 1 \right] - \frac{V + IR_s}{R_{sh}} \quad (64)$$

6.1.3. Photovoltaic module model

Comprising essentially multiple diodes connected in series and parallel, the equivalent circuit of the photovoltaic (PV) module model is

Table 13
The optimal RMSE values and the corresponding estimated parameter values for PV module model.

Algorithm	$I_{ph}(A)$	$I_s(A)$	$R_s(\Omega)$	a	RMSE
L-SHADE	1.03051429913059E+00	3.48226281434371E-06	1.2012710175690E+00	9.811982214248882E+02	2.42507486809496E-03
EBO-with-CMAR	1.03051429883646E+00	3.48226299256927E-06	1.20127100976464E+00	9.811982265416259E+02	2.42507486809497E-03
L-SHADE-SPACMA	1.2579777185085E+00	4.81038604384947E-05	6.57687998814563E-01	7.19290508897137E-01	4.66670181214190E+01
HSES	1.0796983800438E+01	4.42283493905460E-05	9.50039461849769E-01	7.72064073377088E+01	2.13640542777088E+01
E-L-SHADE-SPACMA	1.03051429917001E+00	3.48226290173013E-06	1.20127100944655E+00	9.811982235268765E+02	4.86428975640472E+01
IMODE	1.03051429903366E+00	3.4822626231195614E-06	1.20127102467211E+00	9.811982156232454E+02	4.86428969123858E+01
AGSK	1.030503329499654E+00	3.45622773433084E-06	1.20216402715127E+00	9.71788712626484E+02	4.861139485760203E+01
EAaelig	1.03051429750977E+00	3.48226323802802E-06	1.20127099923886E+00	9.811982445108026E+02	4.8642897302199E+01
IDE-EDA	1.0305142984158E+00	3.4822630911471E-06	1.20127100422845E+00	9.81198233939051E-02	4.8642897720754E+01
APM-JSO	1.03051429838434E+00	3.48226312753028E-06	1.20127100213221E+00	9.811982348833074E+02	4.86428978106567E+01
AL-SHADE	1.03051429855803E+00	3.48226276373739E-06	1.20127101672177E+00	9.8119823039822701E-02	4.86428974075942E+01
mL-SHADE-RL	1.03051429851580E+00	3.48226286569475E-06	1.20127100665141E+00	9.8119822633956286E+02	4.86428975255628E+01
mL-SHADE-SPACMA	8.34604894199690E-01	4.8665560255940E-06	1.43100760943769E-01	7.2360914234240E+01	2.42507486809499E-03
ECL-SHADE-SPACMA	1.03051429943485E+00	3.48226277427635E-06	1.20127101235918E+00	9.811982190332210E-02	2.42507486809496E-03

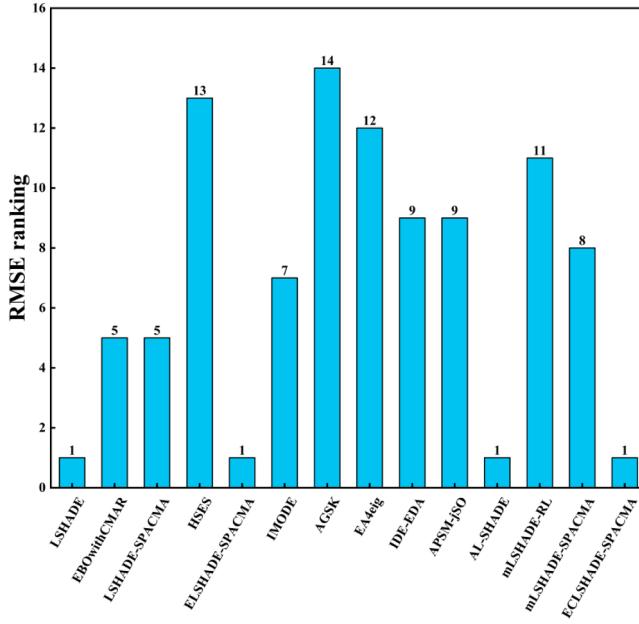


Fig. 26. RMSE-based performance evaluation and ranking of algorithms for PV module model.

illustrated in Fig. 23(c). This model extends the SDM to configurations involving N_s diodes connected in series and N_p diodes in parallel, where N_s denotes the number of series-connected diodes and N_p represents the number of parallel-connected diodes. The module-level output current I is derived as follows:

$$I = I_{ph}N_p - I_sN_p \left[\exp\left(\frac{(VN_p + IR_sN_s) \cdot q}{aN_sN_p kT}\right) - 1 \right] - \frac{VN_p + IR_sN_s}{R_{sh}N_s} \quad (65)$$

For SDM:

$$\begin{cases} f(V, I, \mathbf{X}) = I_{ph} - I_s \left[\exp\left(\frac{(V + IR_s) \cdot q}{akT}\right) - 1 \right] - \frac{+VIR_s}{R_{sh}} - I \\ \mathbf{X} = [I_{ph}, I_s, R_s, R_{sh}, a] \end{cases} \quad (67)$$

For DDM:

$$\begin{cases} f(V, I, \mathbf{X}) = I_{ph} - I_{s1} \left[\exp\left(\frac{(V + IR_s) \cdot q}{a_1 kT}\right) - 1 \right] - I_{s2} \left[\exp\left(\frac{(V + IR_s) \cdot q}{a_2 kT}\right) - 1 \right] - \frac{V + IR_s}{R_{sh}} - I \\ \mathbf{X} = [I_{ph}, I_{s1}, I_{s2}, R_s, R_{sh}, a_1, a_2] \end{cases} \quad (68)$$

6.1.4. Optimization of the objective function for photovoltaic modeling

When applying heuristic algorithms for parameter extraction in photovoltaic models, it is essential to select an appropriate objective function that accurately quantifies the error associated with parameter identification. Commonly used error metrics include absolute error (AE), mean absolute error (MAE), and root mean square error (RMSE). Among these, RMSE provides a more reliable measure of the discrepancy between experimentally observed and model-predicted values. Therefore, RMSE is adopted as the objective function for optimization. The corresponding mathematical formulation is presented as follows:

$$\min_{\mathbf{x}} \text{RMSE}(\mathbf{x}) = \sqrt{\frac{1}{N} \sum_{i=1}^N (I_i - \hat{I}_i)^2} = \sqrt{\frac{1}{N} \sum_{i=1}^N f(V_i, I_i, \mathbf{X})^2} \quad (66)$$

In the Equation, \hat{I}_i denotes the i -th measured current value. A smaller RMSE value indicates a closer agreement between the predicted and measured current values, thereby reflecting higher precision in parameter estimation. Subsequently, for different photovoltaic models, the objective functions and variables can be defined as follows:

For PV module model:

$$\begin{cases} f(V, I, \mathbf{X}) = I_{ph}N_p - I_sN_p \left[\exp\left(\frac{(VN_p + IR_sN_s) \cdot q}{aN_p kT}\right) - 1 \right] - \frac{VN_p + IR_sN_s}{R_{sh}N_s} - I \\ \mathbf{X} = [I_{ph}, I_s, R_s, R_{sh}, a] \end{cases} \quad (69)$$

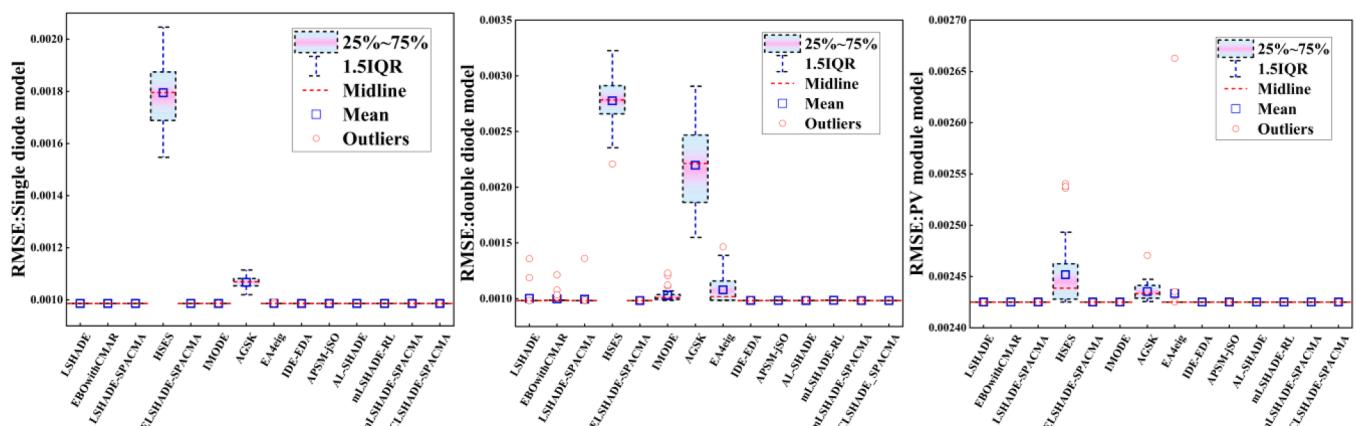
6.2. Experimental results and analysis

This section presents a rigorous evaluation of the proposed ECL-SHADE-SPACMA algorithm in estimating parameters for three photovoltaic models: the SDM, DDM, and PV module model. The performance of ECL-SHADE-SPACMA is compared against 13 advanced algorithms: L-SHADE, EBOwithCMAR, L-SHADE-SPACMA, HSES, ELSHADE-SPACMA, IMODE, AGSK, EA4eig, IDE-EDA, APSM-jSO, AL-SHADE, mLSHADE-RL, and mLSHADE-SPACMA. To ensure fair comparison conditions, each algorithm was allocated the same maximum function evaluation budget ($NFE_{max} = 50000$) for every problem instance. Additionally, each algorithm was independently executed 30

Table 14

Statistical results of RMSE for different algorithms across three models.

Model	Algorithm	RMSE			
		Min	Mean	Max	Std
Single diode model	LSHADE	9.86021877891504E-04	9.86021877891546E-04	9.86021877891585E-04	2.30190613009585E-17
	EBOwithCMAR	9.86021877891492E-04	9.86021877891557E-04	9.86021877891609E-04	2.82772931206508E-17
	LSHADE-SPACMA	9.86021877891478E-04	9.86021877891532E-04	9.86021877891572E-04	2.04842491556201E-17
	HSES	1.54690014652660E-03	1.7947531444389E-03	2.04635400573801E-03	1.35114923473854E-04
	ELSHADE-SPACMA	9.86021877891500E-04	9.86021877891556E-04	9.86021877891599E-04	2.64973087825297E-17
	IMODE	9.86021877891553E-04	9.86021877891860E-04	9.86021877897478E-04	1.06418967025759E-15
	AGSK	1.01923008917836E-03	1.06709329624785E-03	1.11415857695593E-03	2.49018359085585E-05
	EA4eig	9.86021877891587E-04	9.86224045771805E-04	9.92070046187549E-04	1.10413536515289E-06
	IDE-EDA	9.86021877891479E-04	9.86021877891535E-04	9.86021877891579E-04	2.54950923681240E-17
	APSM-jSO	9.86021877891473E-04	9.86021877891551E-04	9.86021877891625E-04	2.69295482185081E-17
	AL-SHADE	9.86021877891472E-04	9.86021877891519E-04	9.86021877891576E-04	2.58805870867747E-17
	mLSHADE-RL	9.86021877891572E-04	9.86021877891648E-04	9.86021877891720E-04	3.87922294005364E-17
	mLSHADE-SPACMA	9.86021877891470E-04	9.86021877891519E-04	9.86021877891579E-04	2.13734570915740E-17
	ECLSHADE-SPACMA	9.86021877891469E-04	9.86021877891497E-04	9.86021877891511E-04	1.02125581658577E-17
Double diode model	LSHADE	9.82484881636855E-04	1.00204843407380E-03	1.35813962091231E-03	7.69774538596482E-05
	EBOwithCMAR	9.82500580948834E-04	9.97880771161118E-04	1.21334084886195E-03	4.51066868358039E-05
	LSHADE-SPACMA	9.82484881636891E-04	9.95317535572752E-04	1.36135095053109E-03	6.91370334697597E-05
	HSES	2.20975946140025E-03	2.77539123932920E-03	3.22721840915711E-03	2.07277223203982E-04
	ELSHADE-SPACMA	9.82484881636910E-04	9.82606951277238E-04	9.83659570111473E-04	2.88413123296886E-07
	IMODE	9.84806223741342E-04	1.03090200032939E-03	1.22900088956956E-03	6.16987231871508E-05
	AGSK	1.54926374239434E-03	2.19757667683431E-03	2.90633590964253E-03	3.75752119289945E-04
	EA4eig	9.84090413602830E-04	1.07920287976513E-03	1.46480740555899E-03	1.27813919893803E-04
	IDE-EDA	9.82484881637015E-04	9.83301455472793E-04	9.86051234785353E-04	1.18970738646508E-06
	APSM-jSO	9.82484881637017E-04	9.83451280003011E-04	9.86021877892854E-04	1.16806060858704E-06
	AL-SHADE	9.82484881636879E-04	9.82957110360285E-04	9.86012605039187E-04	8.90120306237831E-07
	mLSHADE-RL	9.83467722328568E-04	9.84836027271220E-04	9.85907901114541E-04	8.62681391401628E-07
	mLSHADE-SPACMA	9.82484881636907E-04	9.82581338442575E-04	9.85316397100143E-04	5.16611352465632E-07
	ECLSHADE-SPACMA	9.82484881636854E-04	9.82493095148148E-04	9.82507548656081E-04	7.56805555737617E-09
PV module model	LSHADE	2.42507486809496E-03	2.42507486809501E-03	2.42507486809507E-03	2.46889630017987E-17
	EBOwithCMAR	2.42507486809497E-03	2.42507486809501E-03	2.42507486809508E-03	2.82074261136226E-17
	LSHADE-SPACMA	2.42507486809497E-03	2.42507486809500E-03	2.42507486809503E-03	1.70403834454779E-17
	HSES	2.42511645462216E-03	2.45180330103200E-03	2.54051398579526E-03	3.45748511601199E-05
	ELSHADE-SPACMA	2.42507486809496E-03	2.42507486809501E-03	2.42507486809503E-03	1.55902272551582E-17
	IMODE	2.42507486809498E-03	2.42507486809512E-03	2.42507486809599E-03	1.74581269408424E-16
	AGSK	2.42552603331856E-03	2.43532344651452E-03	2.47055104906334E-03	9.10192565096609E-06
	EA4eig	2.42507486809505E-03	2.43333938685044E-03	2.66282753788280E-03	4.33831372664062E-05
	IDE-EDA	2.42507486809500E-03	2.42507486809504E-03	2.42507486809507E-03	1.72490045831054E-17
	APSM-jSO	2.42507486809500E-03	2.42507486809504E-03	2.42507486809508E-03	1.88134367530599E-17
	AL-SHADE	2.42507486809496E-03	2.42507486809500E-03	2.42507486809506E-03	2.53512470687298E-17
	mLSHADE-RL	2.42507486809501E-03	2.42507486809507E-03	2.42507486809511E-03	2.63296992549265E-17
	mLSHADE-SPACMA	2.42507486809499E-03	2.42507486809504E-03	2.42507486809509E-03	3.21235917296489E-17
	ECLSHADE-SPACMA	2.42507486809496E-03	2.42507486809500E-03	2.42507486809507E-03	9.97758544078977E-18

**Fig. 27.** Best RMSE distributions of all algorithms over 30 runs across three model configurations.

times per instance to reduce statistical variability in the results.

Algorithmic accuracy was assessed by comparing the best Root Mean Square Error (RMSE) values obtained. Subsequently, a comprehensive statistical analysis was conducted to evaluate the robustness of the algorithms under varying experimental conditions.

6.2.1. Comparative evaluation of extracted parameter results

This section presents a comprehensive comparative analysis of the extracted parameters, employing the minimal RMSE values obtained across the three photovoltaic models. The optimal results for each experiment are highlighted in bold.

(1) Parameter extraction for SDM

Table 11 summarizes the lowest RMSE values achieved by all algorithms for the single diode model, together with their associated parameter estimates. **Fig. 24** graphically depicts the relative performance ranking of the proposed method and the benchmarked algorithms, based on these minimal RMSE values. Experimental results demonstrate that ECLSHADE-SPACMA achieves the best RMSE, followed in order by mLSHADE-SPACMA, AL-SHADE, APSM-jSO, LSHADE-SPACMA, IDE-EDA, EBOwithCMAR, ELSHADE-SPACMA, LSHADE, IMODE, mLSHADE-RL, EA4eig, AGSK, and HSES. Although the RMSE values of most competing algorithms are statistically comparable to that of ECLSHADE-SPACMA (with the exception of AGSK and HSES), the actual parameters of photovoltaic systems are generally unknown in real-world applications. Therefore, even minor improvements in the objective function can carry substantial practical implications for the optimization of photovoltaic systems.

(2) Parameter extraction for DDM

The determination of seven parameters is required for the DDM. Compared to the SDM, the DDM involves the estimation of two additional parameters, significantly increasing the complexity of the problem and thereby providing a more rigorous test environment for the ECLSHADE-SPACMA algorithm. **Table 12** presents the parameter estimates obtained by various algorithms, along with their corresponding optimal RMSE values. Additionally, **Fig. 25** graphically illustrates the relative ranking based on these optimal RMSE values, offering a clear comparative evaluation of the proposed algorithm against established alternatives. Experimental results indicate that ECLSHADE-SPACMA achieves the lowest RMSE, followed by LSHADE, AL-SHADE, LSHADE-SPACMA, mLSHADE-SPACMA, ELSHADE-SPACMA, IDE-EDA, APSM-jSO, EBOwithCMAR, mLSHADE-RL, EA4eig, IMODE, AGSK, and HSES. It is worth noting that certain algorithms, such as mLSHADE-SPACMA and APSM-jSO, demonstrate strong performance in SDM parameter extraction but exhibit reduced effectiveness in the DDM, suggesting that the number of parameters may impact algorithmic convergence. However, ECLSHADE-SPACMA demonstrates the best performance in both SDM and DDM problems, fully demonstrating its superiority over other mainstream algorithms in dealing with complex optimization problems and thus showing stronger robustness and competitiveness in different photovoltaic model parameter extraction tasks.

(3) Parameter extraction for PV module model

In the photovoltaic module model, five key parameters must be determined. **Table 13** presents the optimal RMSE values along with the corresponding identified parameters, while **Fig. 26** visually illustrates the relative performance rankings of the proposed algorithm and other evaluated algorithms at these optimal RMSE values. Experimental results indicate that ECLSHADE-SPACMA, LSHADE, AL-SHADE, and ELSHADE-SPACMA achieve the most competitive performance. These are closely followed by EBOwithCMAR and LSHADE-SPACMA, which demonstrate comparable efficacy. The subsequent rankings include:

IMODE and mLSHADE-SPACMA; IDE-EDA and APSM-jSO, which exhibit similar performance; and finally, mLSHADE-RL, EA4eig, HSES, and AGSK.

6.2.2. Comparison of experimental results across three PV models

Previous studies primarily assessed algorithm performance using parameter estimation values and RMSE. This section enhances the evaluation framework by incorporating multiple statistical indicators—minimum, maximum, mean, and standard deviation—calculated from 30 independent runs (refer to **Table 14**). This approach enables a more comprehensive quantitative assessment of each algorithm's accuracy and consistency in photovoltaic parameter estimation. ECLSHADE-SPACMA exhibited consistently superior performance across all three photovoltaic models (SDM, DDM, and PV module model), achieving the lowest RMSE mean, the smallest RMSE minimum, and the best RMSE maximum. Moreover, it achieved the lowest standard deviation, reflecting greater reliability and reduced variability in the results. **Fig. 27** illustrates the RMSE distribution characteristics through box plots, depicting central tendency, dispersion, and outlier presence. The algorithm maintained the lowest RMSE median across all models with minimal outliers, indicating high stability. This consistent performance underscores its practical applicability in photovoltaic parameter extraction tasks that demand predictability and robustness.

In summary, the ECLSHADE-SPACMA algorithm proposed in this study exhibits superior solution accuracy and enhanced operational reliability in addressing parameter identification challenges across diverse photovoltaic systems.

7. Summary and future work

7.1. Summary

This paper introduces an enhanced algorithm, ECLSHADE-SPACMA, which represents an improved variant of LSHADE-SPACMA. The proposed algorithm integrates four key components: a dynamic fitness-directed selective pressure mutation strategy, a nonlinear population size reduction mechanism based on an exponential function, an elastic weighted geometric archive pruning strategy, and an adaptive probabilistic local search mechanism driven by success rate feedback. Collectively, these enhancements significantly improve the balance between global exploration and local exploitation. Experimental evaluations on the CEC2014 and CEC2017 benchmark function sets indicate that ECLSHADE-SPACMA significantly outperforms the original LSHADE-SPACMA as well as other mainstream comparative algorithms. Furthermore, the algorithm demonstrates strong performance in practical applications such as UAV path planning in complex mountainous terrains and parameter identification for solar photovoltaic models.

7.2. Future work

The performance of ECLSHADE-SPACMA in low-dimensional optimization problems is suboptimal, primarily due to a mismatch between its sophisticated algorithmic mechanisms and the structural characteristics of low-dimensional search spaces. While features such as nonlinear population size reduction and elastic archive pruning effectively enhance population diversity and mitigate premature convergence in high-dimensional settings, they may introduce unnecessary computational overhead and excessive exploration in lower-dimensional contexts, thereby impairing convergence efficiency. As a result, simplified strategies may be more effective for low-dimensional problems. In contrast, ECLSHADE-SPACMA demonstrates superior performance in high-dimensional optimization tasks compared to established benchmark algorithms. Future research will therefore aim to improve its computational efficiency in low-dimensional scenarios while expanding its applicability to large-scale optimization challenges. A key focus will be on investigating the trade-off between algorithmic complexity and

problem dimensionality, with particular emphasis on adaptive mechanisms that balance exploration and exploitation across different scales. Furthermore, integrating machine learning techniques to enhance predictive modeling and exploring the application of ECLSHADE-SPACMA in diverse domains—such as medical diagnostics, financial modeling, and feature selection—will constitute important directions for future investigation.

CRediT authorship contribution statement

Wenhao Mao: Writing – review & editing, Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Shengwei Fu:** Investigation, Formal analysis, Data curation, Validation. **Guozhang Zhang:** Investigation, Visualization, Writing – review & editing. **Haisong Huang:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Guangming Gong:** Software, Resources, Formal analysis. **Yulu Liu:** Data curation, Visualization, Writing – original draft. **Honghai Fu:** Resources, Investigation. **Yinwei Li:** Methodology, Validation. **Yongpeng Zhao:** Data curation, Software. **Langlang Zhang:** Formal analysis, Writing – review & editing. **Jiawei Wang:** Resources, Supervision.

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 by the National Natural Science Foundation of China (Nos. 52165063 and 52575566), and the Guizhou Province Science and Technology Projects (QKHC-LH [2024] ZD No. 017, QKHC-DXGA [2025] No. 002, QKHZC-DXGA [2025] No. 008, QKHZC-DXGA [2025] No. 018, QKHZC [2024] No. 093, QKHPTRC-CXTD [2023] No. 007, QKHZC [2023] No. 348, QKHZC [2023] No. 309, QKHPTRC-GCC [2022] No. 066-1).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at [doi:10.1016/j.swevo.2025.102257](https://doi.org/10.1016/j.swevo.2025.102257).

Data availability

Data will be made available on request.

References

- [1] Yingchao Dong, Shaohua Zhang, Hongli Zhang, Xiaojun Zhou, Jiading Jiang, Chaotic evolution optimization: A novel metaheuristic algorithm inspired by chaotic dynamics, *Chaos, Solit. Fract.* 192 (2025).
- [2] Rainer Storn, Kenneth Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *J. Glob. Optim.* 11 (1997).
- [3] Lianhai Lin, Liqin Tian, Zhigang Wang, Wenguang Yang, A differential evolution-based single-level algorithm for jointly optimizing the deployment and flight trajectory of UAV-assisted data collection system, *Appl. Soft. Comput.* (2025).
- [4] Charaf Chermite, Moulay Rachid Douiri, Differential evolution algorithm featuring novel mutation combined with Newton-Raphson method for enhanced photovoltaic parameter extraction, *Energy Convers. Manag.* 326 (2025).
- [5] Shengwei Fu, Ke Li, Haisong Huang, Yongpeng Zhao, Enhanced Iteration Closest Point is Based on Stochastic Differential Evolution Algorithm and Random Sampling, *IEEE*, City, 2024.
- [6] Peixin Yang, Zhongbo Hu, Yang Zhou, Qinghua Su, Wentao Xiong, Grey prediction evolution algorithm with a dominator guidance strategy for solving multi-level image thresholding, *Appl. Soft. Comput.* 174 (2025).
- [7] Fengling Peng, Xing Chen, Jingkai Xue, AI-Assisted antenna optimization: Integrating evolutionary and inverse cascade neural networks with differential evolution, *IEEE Trans. Antennas. Propag.* (2025).
- [8] Jiaqian Yang, Henrique Buglia, Mindaugas Jarmolovičius, Romulo Aparecido, Eric Sillekens, Ronit Sohanpal, Mingming Tan, Dini Pratiwi, Ruben S Luis, Benjamin J Puttnam, 122.6 Tb/s S+ C+ L band unrepeated transmission over 223 km link with optimized bidirectional Raman amplification, *J. Lightwave Technol.* 43 (2025) 4.
- [9] Yu Bai, Pei-Fa Sun, Tian-Hong Wang, Bing Sun, Wei-Jie Yu, Jing-Hui Zhong, Guo-Huan Song, Sang-Woon Jeon, Sam Tak Wu Kwong, Jun Zhang, UAV path planning for data collection from wireless sensor network with matrix-based evolutionary computation, *IEEE Trans. Intell. Transp. Syst.* (2025).
- [10] Yaoyao He, Xiaoyu Hong, Ning Xian, Long-term scheduling strategy of hydro-wind-solar complementary system based on chaotic elite selection differential evolution algorithm with death penalty function, *Eng. Appl. Artif. Intell.* 142 (2025).
- [11] Shengwei Fu, Haisong Huang, Chi Ma, Jianan Wei, Yiting Li, Youfa Fu, Improved dwarf mongoose optimization algorithm using novel nonlinear control and exploration strategies, *Expert. Syst. Appl.* 233 (2023).
- [12] Minyang Chen, Chenchen Feng, Ran Cheng, Metade: evolving differential evolution by differential evolution, *IEEE Trans. Evolut. Computat.* (2025).
- [13] Jingqiao Zhang, Arthur C Sanderson, JADE: adaptive differential evolution with optional external archive, *IEEE Trans. Evolut. Computat.* 13 (2009) 5.
- [14] Ryoji Tanabe, Alex Fukunaga, Success-History Based Parameter Adaptation for Differential Evolution, *IEEE*, City, 2013.
- [15] Ryoji Tanabe, Alex S Fukunaga, Improving the Search Performance of SHADE using Linear Population Size Reduction, *IEEE*, City, 2014.
- [16] Ali W Mohamed, Anas A Hadi, Anas M Fattouh, Kamal M Jambi, LSHADE with Semi-Parameter Adaptation Hybrid with CMA-ES for Solving CEC 2017 Benchmark Problems, *IEEE*, City, 2017.
- [17] Nikolas Hansen, Sibylle D Müller, Petros Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), *Evol. Comput.* 11 (2003) 1.
- [18] Anas A Hadi, Ali W Mohamed, Kamal M Jambi, Single-objective real-parameter optimization: Enhanced LSHADE-SPACMA algorithm. *Heuristics for Optimization and Learning*, 2021.
- [19] Ali Wagdy Mohamed, Ali Khater Mohamed, Adaptive guided differential evolution algorithm with novel mutation for numerical optimization, *Int. J. Mach. Learn. Cybern.* 10 (2019).
- [20] Ali Wagdy Mohamed, Anas A Hadi, Ali Khater Mohamed, Differential evolution mutations: taxonomy, comparison and convergence analysis, *IEEE Access* 9 (2021).
- [21] Vladimir Stanovov, Shakhnaz Akhmedova, Eugene Semenkin, LSHADE Algorithm with Rank-Based Selective Pressure Strategy for Solving CEC 2017 Benchmark Problems, *IEEE*, City, 2018.
- [22] Yintong Li, Tong Han, Shangqin Tang, Changqiang Huang, Huan Zhou, Yuan Wang, An improved differential evolution by hybridizing with estimation-of-distribution algorithm, *Inf. Sci.* 619 (2023).
- [23] Yintong Li, Tong Han, Huan Zhou, Yujie Wei, Yuan Wang, Mulai Tan, Changqiang Huang, APSM-jSO: A novel jSO variant with an adaptive parameter selection mechanism and a new external archive updating mechanism, *Swarm. Evol. Comput.* 78 (2023).
- [24] Yintong Li, Tong Han, Xiaofei Wang, Huan Zhou, Shangqin Tang, Changqiang Huang, Bo Han, MJSO: A modified differential evolution with a probability selection mechanism and a directed mutation strategy, *Swarm. Evol. Comput.* 78 (2023).
- [25] Quanbin Zhang, Zhenyu Meng, HAPI-DE: Differential evolution with hierarchical archive based mutation strategy and promising information, *Swarm. Evol. Comput.* 91 (2024).
- [26] Juncan Li, Zhenyu Meng, Global opposition learning and diversity enhancement based differential evolution with exponential crossover for numerical optimization, *Swarm. Evol. Comput.* 87 (2024).
- [27] Yongjun Sun, Tingting Sun, Zujun Liu, A framework of integrated differential evolution variants based on adaptive relay mode for global optimization, *Appl. Soft. Comput.* 167 (2024).
- [28] Xiaolin Yi, Xianfeng Ding, Qian Chen, A differential evolution algorithm considering multi-population based on birth & death process and opposition-based learning with condition, *Swarm. Evol. Comput.* 96 (2025).
- [29] Guanyu Yuan, Gaoji Sun, Libao Deng, Chunlei Li, Guoqing Yang, Lili Zhang, A periodic intervention and strategic collaboration mechanisms based differential evolution algorithm for global optimization, *Appl. Soft. Comput.* (2025).
- [30] Lun Zhu, Yongquan Zhou, Guo Zhou, Qifang Luo, Yuanfei Wei, Solving spherical multi-aircraft path planning problem using self-adaptive update strategy differential evolution algorithm, *Swarm. Evol. Comput.* 96 (2025).
- [31] Rui Zhong, Zhongmin Wang, Yujun Zhang, Junbo Jacob Lian, Jun Yu, Huijing Chen, Integrating Competitive Framework into Differential Evolution: Comprehensive performance analysis and application in brain tumor detection, *Appl. Soft. Comput.* 174 (2025).
- [32] Bozhen Chen, Haibin Ouyang, Steven Li, Weiping Ding, Dual-stage self-adaptive differential evolution with complementary and ensemble mutation strategies, *Swarm. Evol. Comput.* 93 (2025).
- [33] Mengnan Tian, Xingbao Gao, Xueqing Yan, Performance-driven adaptive differential evolution with neighborhood topology for numerical optimization, *Knowl. Based. Syst.* 188 (2020).
- [34] Ali Wagdy Mohamed, Hattan F Abutarboush, Anas A Hadi, Ali Khater Mohamed, Gaining-sharing knowledge based algorithm with adaptive parameters for engineering optimization, *IEEE Access* 9 (2021).
- [35] Guozhang Zhang, Shengwei Fu, Ke Li, Haisong Huang, Differential evolution with multi-strategies for UAV trajectory planning and point cloud registration, *Appl. Soft. Comput.* 167 (2024).
- [36] Jing J Liang, Bo Y Qu, Ponnuthurai N Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single

- objective real-parameter numerical optimization, *Computat. Intell. Lab., Zhengzhou Univ., Zhengzhou China Techn. Rep., Nanyang Technol. Univ., Singapore* 635 (2013) 2.
- [37] Guohua Wu, Rammohan Mallipeddi, Ponnuthurai Suganthan, *Tech. Rep. Nanyang Technol. Univ., Singapore*, 2016.
- [38] Abhishek Kumar, Rakesh Kumar Misra, Devender Singh, Improving the Local Search Capability of Effective Butterfly Optimizer Using Covariance Matrix Adapted Retreat Phase, *IEEE, City*, 2017.
- [39] Geng Zhang, Yuhui Shi, Hybrid Sampling Evolution Strategy for Solving Single Objective Bound Constrained Problems, *IEEE, City*, 2018.
- [40] Karam M Sallam, Saber M Elsayed, Ripon K Chakrabortty, Michael J Ryan, Improved Multi-Operator Differential Evolution Algorithm for Solving Unconstrained Problems, *IEEE, City*, 2020.
- [41] Ali Wagdy Mohamed, Anas A Hadi, Ali Khater Mohamed, Noor H Awad, Evaluating The Performance of Adaptive Gainingsharing Knowledge Based Algorithm on CEC 2020 Benchmark Problems, *IEEE, City*, 2020.
- [42] Peter Bujok, Patrik Kolenovsky, Eigen Crossover In Cooperative Model of Evolutionary Algorithms Applied to CEC 2022 Single Objective Numerical Optimisation, *IEEE, City*, 2022.
- [43] Yintong Li, Tong Han, Huan Zhou, Shangqin Tang, Hui Zhao, A novel adaptive L-SHADE algorithm and its application in UAV swarm resource configuration problem, *Inf. Sci.* 606 (2022).
- [44] Dikshit Chauhan, and Anupam Trivedi. 2024. A multi-operator ensemble LSHADE with restart and local search mechanisms for single-objective optimization. arXiv preprint arXiv:2409.15994.
- [45] Shengwei Fu, Chi Ma, Ke Li, Cankun Xie, Qingsong Fan, Haisong Huang, Jiangxue Xie, Guozhang Zhang, Mingyang Yu, Modified LSHADE-SPACMA with new mutation strategy and external archive mechanism for numerical optimization and point cloud registration, *Artif. Intell. Rev.* 58 (2025) 3.
- [46] Joaquín Derrac, Salvador García, Daniel Molina, Francisco Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm. Evol. Comput.* 1 (2011) 1.
- [47] Milton Friedman, A comparison of alternative tests of significance for the problem of m rankings, *Ann. Math. Stat.* 11 (1940) 1.
- [48] Zhaolong Ning, Tengfeng Li, Yu Wu, Xiaojie Wang, Qingqing Wu, Fei Richard Yu, Song Guo, 6G communication new paradigm: the integration of unmanned aerial vehicles and intelligent reflecting surfaces, *IEEE Commun. Surv. Tutor.* (2025).
- [49] Ihsan Sadati, An efficient matheuristic integration with benders decomposition for unmanned aerial vehicle routing problem in forest fire surveillance, *Comput. Ind. Eng.* 204 (2025).
- [50] Yaowei Li, Chen Zhang, Weifeng Su, Shaojie Jiang, Dongyang Nie, Yaying Wang, Yuzheng Wang, Hongdi He, Qi Chen, Scot T Martin, Copter-type UAV-based sensing in atmospheric chemistry: recent advances, applications, and future perspectives, *Environ. Sci. Technol.* (2025).
- [51] Xue Fu, Yu Wang, Yun Lin, Tomoaki Ohtsuki, Bamidele Adebisi, Guan Gui, Hikmet Sari, Towards collaborative and cross-environment UAV classification: federated semantic regularization, *IEEE Trans. Inf. Forens. Secur.* (2025).
- [52] Jianhua Tang, Yao Zeng, UAV data acquisition and processing assisted by UGV-enabled mobile edge computing, *IEEE Trans. Industr. Inform.* (2025).
- [53] Dashuai Wang, Minghu Zhao, Zhiqin Li, Sheng Xu, Xiaohu Wu, Xuan Ma, Xiaoguang Liu, A survey of unmanned aerial vehicles and deep learning in precision agriculture, *Euro. J. Agron.* 164 (2025).
- [54] Shafiq Akter, Dat Van Anh Duong, Seokhoon Yoon, Joint optimization of UAV trajectory, task offloading, and resource allocation in UAV-aided emergency response operations, *IEEE Internet. Things. J.* (2025).
- [55] Hong Wang, Chenyang Zhao, Yuan Feng, Xu Huang, Chenyi Qu, Yaguang Zhu, Ming Xin, Wenbin Guo, Enhancing firefighter safety and efficiency through UAV-assisted AI-based human motion recognition system, *Expert. Syst. Appl.* (2025).
- [56] Junhong Jin, Genlai Zhang, Xin Li, Xichao Su, Chen Lu, Yujie Cheng, Yu Ding, Lei Wang, Xinwei Wang, Cross-platform mission planning for UAVs under carrier delivery mode, *Def. Technol.* (2025).
- [57] Yuheng Yang, Xing Guo, Xingshuo Hai, Qiang Feng, Bo Sun, Zili Wang, Modeling and vulnerability analysis of UAV swarm based on two-layer multi-edge complex network, *Reliab. Eng. Syst. Saf.* 256 (2025).
- [58] Kiri van den Wall Bake, Aschalew Tigabu, Marta Talevi, Pieter van Beukering, Marije Schaafsma, Solar PV and clean cookstove technology diffusion systems: Four case studies from Sub-Saharan Africa, *Renew. Energy* 240 (2025).
- [59] Yanwei Sun, Ying Li, Run Wang, Renfeng Ma, Dynamic geospatial modeling of solar energy expansion potential in China: Implications for national-level optimization of solar photovoltaic plant layouts, *Energy* (2025).
- [60] Fahim Ullah, Kamran Hasrat, Sami Iqbal, Sunel Kumar, Shuang Wang, Mao Mu, Wanning Lu, Evaluating the global warming potential of a 4.6 kWp solar PV system in Karak-KPK: a life cycle assessment, *Appl. Therm. Eng.* 266 (2025).
- [61] Haoyu Wang, Xiaobing Yu, Yangchen Lu, A reinforcement learning-based ranking teaching-learning-based optimization algorithm for parameters estimation of photovoltaic models, *Swarm. Evol. Comput.* 93 (2025).
- [62] T.R. Ayodele, A.S.O. Ogunjuyigbe, E.E. Ekoh, Evaluation of numerical algorithms used in extracting the parameters of a single-diode photovoltaic model, *Sustain. Energy Technol. Assess.* 13 (2016).
- [63] T. Sudhakar Babu, J. Prasanth Ram, K. Sangeetha, Antonino Laudani, N. Rajasekar, Parameter extraction of two diode solar PV model using Fireworks algorithm, *Solar Energy* 140 (2016).
- [64] M.R. AlRashidi, M.F. AlHajri, K.M. El-Naggar, A.K. Al-Othman, A new estimation approach for determining the I-V characteristics of solar cells, *Solar Energy* 85 (2011) 7.