

Article

Knowledge-Based Perturbation LaF-CMA-ES for Multimodal Optimization

Huan Liu ^{1,2,*} , Lijing Qin ^{1,2} and Zhao Zhou ²
¹ School of Computer and Communication Technology, Lanzhou University of Technology, Lanzhou 730050, China; qinlij@gsau.edu.cn

² College of Information Science and Technology, Gansu Agricultural University, Lanzhou 730070, China; zzzzzz20230206@163.com

* Correspondence: liuh1204@hotmail.com

Abstract: Multimodal optimization presents a significant challenge in optimization problems due to the existence of multiple attraction basins. Balancing exploration and exploitation is essential for the efficiency of algorithms designed to solve these problems. In this paper, we propose the Kbp-LaF-CMAES algorithm to address multimodal optimization problems based on the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) framework. The Leaders and Followers (LaF) and Knowledge-based Perturbation (KbP) strategies are the primary components of the Kbp-LaF-CMAES algorithm. The LaF strategy is utilized to extensively explore the potential local spaces, where two cooperative populations evolve in synergy. The KbP strategy is employed to enhance exploration capabilities. Improved variants of CMA-ES are used to exploit specific domains containing local optima, thereby potentially identifying the global optimum. Simulation results on the test suite demonstrate that Kbp-LaF-CMAES significantly outperforms other meta-heuristic algorithms.

Keywords: multimodal optimization; leaders and followers; covariance matrix adaptation evolution strategy; knowledge-based perturbation



Citation: Liu, H.; Qin, L.; Zhou, Z. Knowledge-Based Perturbation LaF-CMA-ES for Multimodal Optimization. *Appl. Sci.* **2024**, *14*, 9133. <https://doi.org/10.3390/app14199133>

Academic Editor: Paolino Di Felice

Received: 26 June 2024

Revised: 6 August 2024

Accepted: 13 August 2024

Published: 9 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multimodal continuous optimization (MMOP) is particularly challenging due to the presence of multiple attraction basins, each containing a local optimum. In these landscapes, solutions within a specific basin quickly converge to the local optimum through a local greedy search [1,2]. However, the presence of multiple local optima complicates the direct identification of the global optimum using local search strategies.

In real-world scenarios, optimal results are often unattainable due to physical constraints, such as limited time [3]. Some methods have been introduced to address MMOP [4]. For single-objective continuous optimization problems, evolutionary algorithms (EAs) typically search for one optimal solution. However, for MMOPs, numerous local optimal solutions must be identified to provide different perspectives to the user [5]. Novel strategies, including niching methods and population decomposition, have been introduced to tackle MMOPs.

Niching techniques divide the entire population into species, and subsequent operators, such as mutation and crossover, are applied based on these species divisions. For example, using affinity propagation clustering (APC), Wang et al. [6] introduced an automatic niching technique and developed a niching differential evolution (DE) algorithm to solve MMOPs. Similarly, Hu et al. [7] introduced a niching backtracking search algorithm with adaptive local search to address MMOPs. Various techniques such as speciation, crowding, fitness sharing, clustering, derating, and dynamic cluster size niching have been employed to enhance the performance of niching-based multimodal algorithms, particularly in low-dimensional MMOPs. However, these multimodal algorithms often suffer

from performance sensitivity due to multiple parameter-dependent strategies and struggle in higher-dimensional scenarios.

The method of multi-objective evolutionary algorithms (MOEAs) is also applied to solve MMOPs [2]. This approach involves constructing objectives for each subproblem to reformulate the optimization problem [8]. Peng introduced a subset selection framework and multi-objective multimodal algorithms [9]. The effectiveness of these algorithms largely depends on the constructed objectives.

The performance of EAs in solving MMOPs often surpasses that of classical methods [10,11]. EAs simulate natural processes by maintaining a population through selection, mutation, and recombination strategies [12]. This evolutionary path approximates the solution's evolutionary trajectory. By extracting hidden information from historical solutions, EAs can uncover problem characteristics and identify the optimum accordingly.

Improved EAs have demonstrated strong performance in certain MMOPs [13]. For example, Wang et al. [14] developed an adaptive distributed differential evolution algorithm for MMOPs. Agrawal [15] introduced a distributed mutation framework along with an elite archive mechanism. Sheng et al. [16] proposed an adaptive neighborhood mutation method based on memetic differential evolution for MMOPs. Additionally, Lin et al. [17] developed an enhanced differential evolution algorithm incorporating strategies such as nearest-better clustering (NBC), species balance, and key point-based mutation for MMOPs.

Despite their capabilities, these algorithms often rely on multiple search procedures, leading to a somewhat random search process. Additionally, their ability to self-learn problem knowledge is limited. Consequently, researchers have been working to develop EAs that can identify multiple global optima and reliably reach the global optimum in MMOPs, thereby enhancing EA performance.

CMA-ES is an effective optimization strategy that is widely utilized across various problems. In the context of MMOPs, CMA-ES incorporates several strategies. Ahrari et al. [18] utilized small populations to explore the problem space in parallel, designating identified basins as taboo fields to prevent other subpopulations from converging on the same solution. Li et al. [19] employed opposition learning to investigate unknown regions. Luo et al. [1] proposed an ensemble nearest-better-neighbor clustering method to ensure that solutions are distributed across different basins.

In this paper, we propose an enhanced Leader-Follower (LaF) approach to minimize the overexploitation phenomenon and introduce a CMA-ES-based MMOP solver. The method combines the adaptive evolutionary properties of CMA-ES, the powerful over-exploitation capability of LaF, and the Knowledge-based Perturbation (KbP) strategy. The algorithm proposed in [20] utilizes some of the algorithmic ideas presented in this paper. However, this paper further proposes a knowledge-driven perturbation-based strategy, which significantly improves the convergence speed and solution accuracy of the algorithm. In addition, this paper not only extends and enriches the experimental content but also adds analyses and discussions not previously covered. Specifically, we incorporate more comprehensive experimental results, refine the methodology, and delve into the implications of the research findings. These improvements significantly enhance the robustness and depth of the study.

The structure of this paper is organized around the algorithm's components. Section 2 is an introductory overview of CMA-ES, the LaF strategy, and the KbP method. Section 3 elaborates on an improved algorithm and details the employed strategies. Section 4 presents simulation results and analysis to demonstrate the algorithm's performance. Finally, the paper concludes in Section 5.

2. Materials and Methods

2.1. Leaders and Followers (LaF)

Multiple attraction basins exist in MMOPs, each varying in size. To effectively solve MMOPs, it is crucial to identify sufficient basins of the function to extract adequate knowledge about the optimization problem. To avoid getting trapped in local optima, some

algorithms address MMOPs by controlling parameters [21]. For solving MMOPs, exploration is more important than exploitation. Overemphasis on exploitation can negatively impact algorithm efficiency [22].

The LaF mechanism is a novel approach designed to address multimodal problems [23]. In the LaF mechanism, there are two sub-populations: leaders and followers. The leader population consists of the optimal solutions identified during the optimization process, while the follower population is guided by these leaders. Figure 1 illustrates the LaF strategy's procedure. In the strategy of LaF, the direct comparison of exploratory solutions with the best-known solutions can be avoided. The leader population is periodically updated with solutions from the follower population over several generations. By repeatedly isolating the accumulated information, the LaF strategy is appropriate for MMOPs.

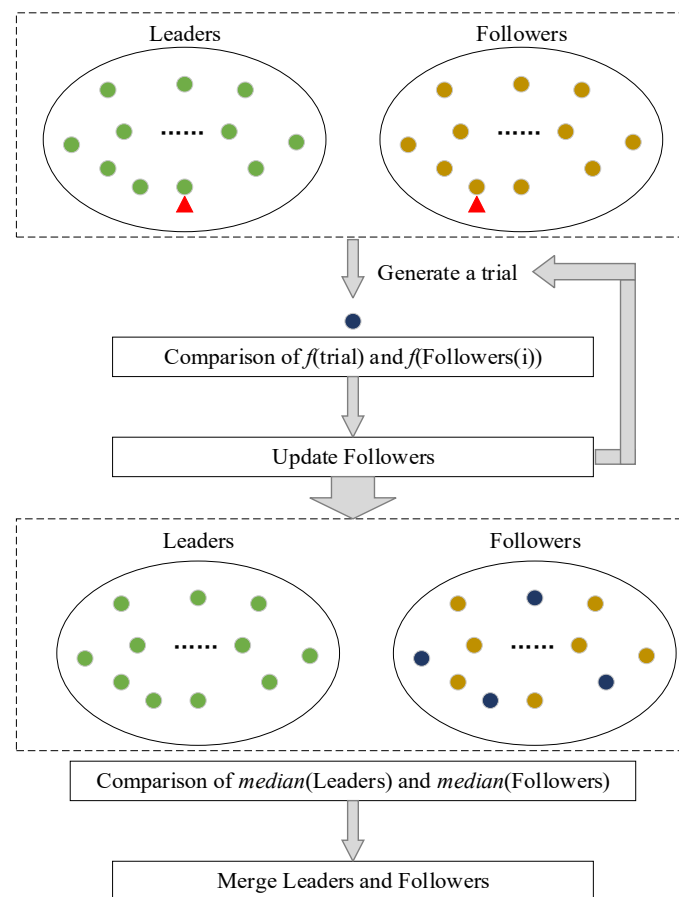


Figure 1. The framework of the LaF strategy.

Firstly, two individuals are selected from the leaders and followers as $leader_i$ and $follower_i$, respectively. Then, a new individual $trial_i$ is generated by recombination.

$$trial_i = leader_i + \varepsilon_i * 2 * (leader_i - follower_i) \quad (1)$$

where ε_i is a random number, the value is chosen from 0 and 1. Then, the solution $trial_i$ is compared with the corresponding solution in the followers, and the individual $trial_i$ with better fitness value is retained. In the updating process of the followers, the offspring solutions are not compared with the current optimal solutions, that is, the individuals in the leaders, but with the individuals of followers. Therefore, the degree of exploitation in the LaF mechanism is relatively low, which improves the success rate of exploration.

2.2. Knowledge-Based Perturbation

The MMOPs are complex and consist of numerous local optima, making it challenging for algorithms to find the global optimum. A restart strategy can temporarily address this issue [24]. However, after restarting, the algorithm may quickly cease to converge. Due to the varying dimensions and complexities of landscapes, the efficiency of restart-based algorithms is often unstable.

The Knowledge-based Perturbation (KbP) mechanism is a perturbation restart method designed to prevent the algorithm from getting trapped in local optima and to continue exploring the problem space [25]. In the perturbation process, it is crucial to determine the size and timing of the perturbations. The framework of the KbP strategy is illustrated in Figure 2.

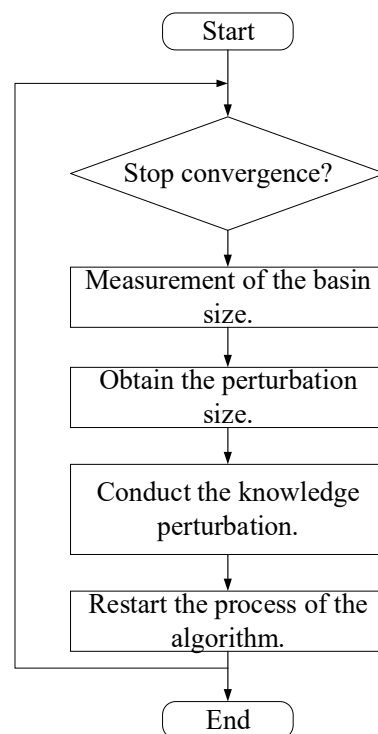


Figure 2. The procedure of the KbP strategy.

If the introduced method is detected as stalled for the optimization problem, the KbP method is triggered. In the KbP strategy, the size of the local basin is used to guide the exploration process. The size of local basins d_{basin} in MMOPs is determined using a linear search mechanism, requiring $1000 * d$ function evaluations, where d is the dimension of the MMOP. Specifically, d line searches are performed, each utilizing 1000 function evaluations. In each line search, one term spans the entire range of the search space, while the other terms vary between random pairs of values. This approach reveals the landscape of local basins for subsequent strategies in the algorithm.

During the perturbation process, the size of each dimension is generated based on d_{basin} . The perturbation is conducted on the Leaders' individuals.

$$leaders_{new} = leaders_{old} + d_{pert} \quad (2)$$

$$d_{pert} = 3/4 * rand(-d_{basin}, d_{basin}) \quad (3)$$

Here, the new individuals in the Leaders' population are generated by combining the original individuals $leaders_{old}$ with d_{pert} .

2.3. CMA-ES

The CMA-ES algorithm, proposed by Hansen & Ostermeier in 1996 [26], is one of the most advanced algorithms for solving continuous optimization problems. In CMA-ES, a multivariate Gaussian distribution model is hypothesized to describe the previous motion pattern of the problem in the search space. During the iterative process, relevant parameters such as the covariance matrix and step size are adjusted based on historical information. New individuals are generated through sampling from the adjusted distribution, and the best individuals are selected to update the parameters. The CMA-ES is invariant to translation and rotation. Therefore, CMA-ES is invariant to translation and rotation, making it highly effective for solving ill-conditioned and complex problems.

The CMA-ES algorithm consists of four main processes: sampling, selection, recombination, and parameter updating. The most common CMA-ES is (μ, λ) -CMA-ES, where μ and λ are the numbers of parent and child populations, respectively. In the $g + 1$ generation of the iterative process of (μ, λ) -CMA-ES, λ individuals are generated by sampling: $X_k^{(g+1)} \sim m^g + \sigma^g N(0, C^g)$, $k = 1, 2, \dots, \lambda$, where m^g is the mean value of individuals after population selection of g generation; σ^g is the standard deviation of g generation, also known as step size; C^g is the covariance matrix and $N(0, C^g)$ is the multivariate normal distribution with the mean value of 0 and covariance matrix of C^g .

Individuals are selected according to their fitness value. The first μ individuals are selected as parents to enter the recombination process. In the process of recombination, $m^{(g+1)}$ is updated as $m^{(g+1)} = \sum_{i=1}^{\mu} \omega_i \cdot x_{i:\lambda}^{(g+1)}$, where ω_i is the weight coefficient of reorganization, $x_{i:\lambda}^{(g+1)}$ denotes the i -th best individual among the λ individuals of the $g + 1$ generation.

$$f(x_{1:\lambda}^{(g+1)}) \leq f(x_{2:\lambda}^{(g+1)}) \leq \dots \leq f(x_{\lambda:\lambda}^{(g+1)}) \quad (4)$$

where $f(x)$ is the fitness of the individual x .

Finally, the parameters of σ^g and C^g are updated. The step size σ is updated according to the evolution path. The evolution path p_σ is updated by a distribution that obeys $N(0, I)$ according to the deviation of the mean value as shown in the following formula.

$$p_\sigma^{(g+1)} = (1 - c_\sigma) p_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma) \mu_{eff} C^{(g)}}^{-\frac{1}{2}} \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \quad (5)$$

where, c_σ is the learning rate of the cumulative evolutionary path and μ_{eff} is the variance effective selection quality.

$$\ln \sigma^{(g+1)} = \ln \sigma^{(g)} + \frac{c_\sigma}{d_\sigma \cdot E\|\mathcal{N}(0, I)\|} \quad (6)$$

where d_σ is the attenuation coefficient and $E\|\mathcal{N}(0, I)\|$ is the expectation of the Euclidean norm of a vector of $\mathcal{N}(0, I)$ distribution.

The update of the covariance matrix C is the core part of the CMA-ES algorithm. C is updated with the following formula.

$$C^{(g+1)} = (1 - c_1 - c_\mu) C^{(g)} + c_1 \left(p_c^{(g+1)} \left(p_c^{(g+1)} \right)^T \right) + c_\mu \sum_{i=1}^{\mu} \omega_i y_{i:\lambda}^{(g+1)} \left(y_{i:\lambda}^{(g+1)} \right)^T \quad (7)$$

where c_1 and c_μ are the learning rates of the rank 1 and rank μ update of the covariance matrix, respectively, and $p_c^{(g+1)}$ is an evolutionary path similar to $p_c^{(g+1)}$.

$$y_{i:\lambda}^{(g+1)} = \frac{x_{i:\lambda}^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \quad (8)$$

$$p_c^{(g+1)} = (1 - c_c)p_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{eff}} \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \quad (9)$$

3. The KbP-LaF-CMAES Algorithm

The KbP-LaF-CMAES algorithm operates in two distinct stages: identifying sufficient attraction basins and exploiting the identified basins. The LaF method is used to explore local basins by leveraging problem-specific knowledge during the initial search, specifically the size of the attraction basin across various dimensions. The improved variants of CMA-ES are introduced for exploitation.

3.1. Bi-Population Cooperative Evolution Strategy

The goal of methods designed for MMOPs is to discover sufficient global and local optima within the optimization problem's landscape. This paper proposes a new method: a bi-population cooperative approach with a KbP strategy. This method leverages two distinct populations that evolve cooperatively, as depicted in Figure 3.

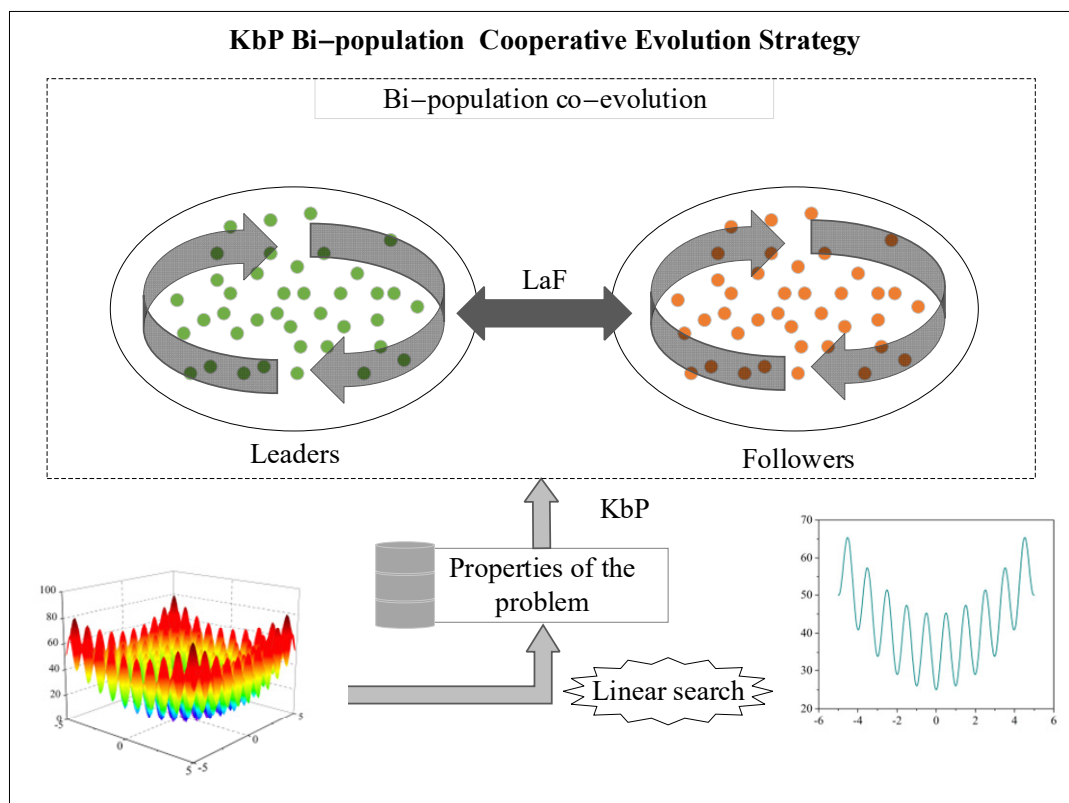


Figure 3. The framework of the bi-population cooperative evolution strategy.

In the bi-population method, individuals are updated using the LaF method, as outlined in Section 2. New solutions are juxtaposed with corresponding solutions in the follower population to maintain population diversity. The LaF strategy mitigates comparison bias associated with elitism and effectively identifies promising basins. In instances of stagnation, perturbations based on multimodal knowledge are implemented.

In the KbP strategy, the population is perturbed using a random value determined by the multimodal characteristics of the problem. Subsequently, the search process of the LaF strategy is restarted from the perturbed solutions. The contribution of the KbP strategy in multimodal problems is illustrated in Figure 4. For convenience, the 2D Rastrigin problem serves as an example to demonstrate the influence of KbP. As shown in Figure 4a, a small perturbation value is added to the original solutions, with the new solution remaining in the same basin. This perturbation does not affect the exploitation process. In Figure 4b,c,

the perturbation value corresponds to the size of the basin, allowing solutions to explore adjacent basins. These figures indicate that the direction of perturbation is crucial for exploration. In Figure 4d, if the perturbation size is too large, the new solution's basin becomes random.

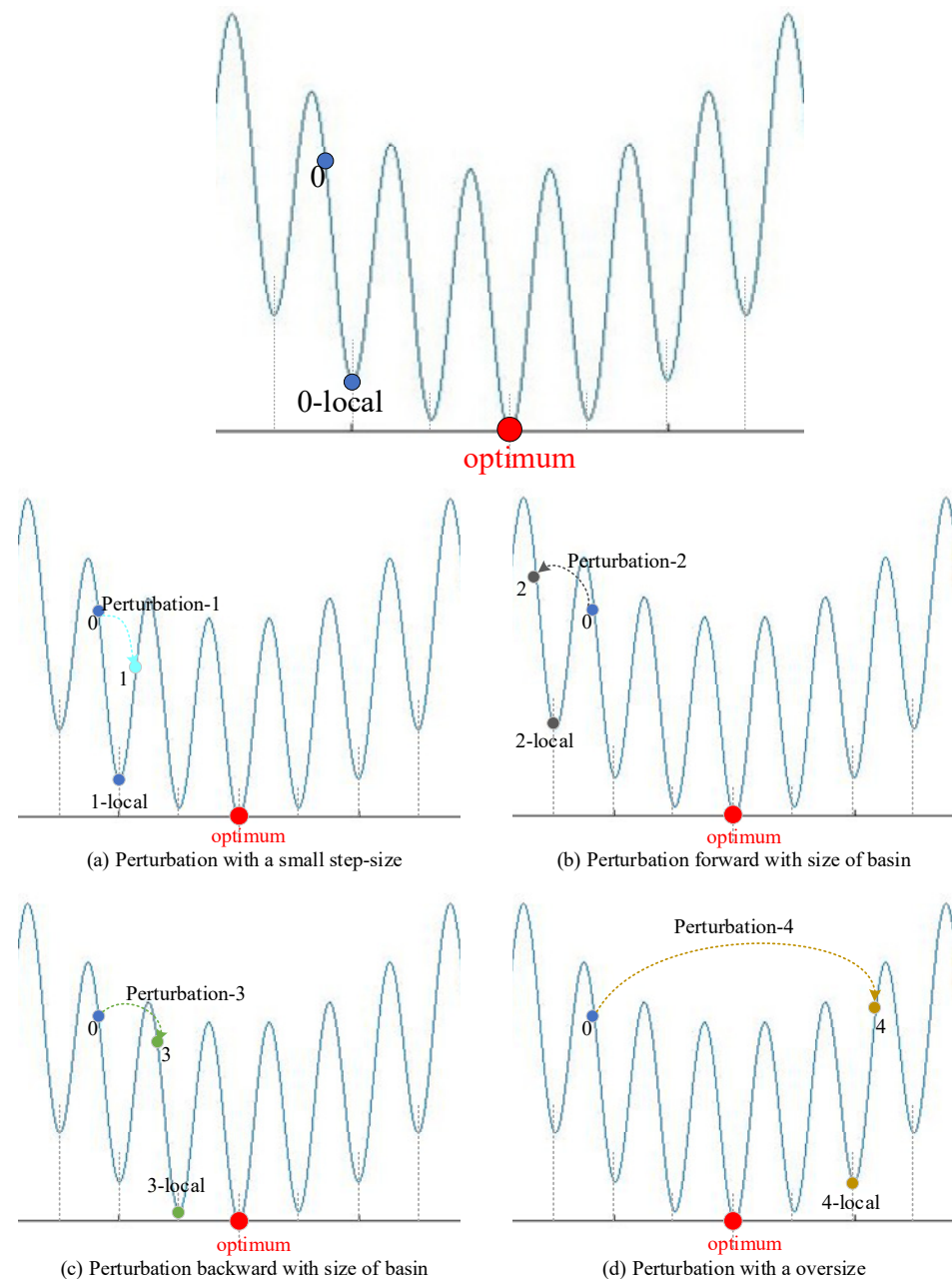


Figure 4. The illustration of the KbP strategy.

In the KbP strategy, the perturbation value is determined by the basin size of the multimodal problem. This knowledge is acquired through the linear search method. A detailed description is provided in Section 2. The minimum basin size in a specific dimension is used to generate the magnitude of the perturbation. During the KbP strategy, solutions are perturbed by a random value related to the basin size.

3.2. CMA-ES with Local Refinement

The CMA-ES is a highly effective algorithm for locating a single global optimum. Therefore, the CMA-ES is employed in the exploitation stage of the proposed algorithm.

In an evolutionary algorithm, the population evolves as a whole, and the population's information is utilized to explore the problem space. However, individual information has not been fully leveraged in this algorithm [27]. This paper proposes the CMA-ES with local refinement (LR-CMA-ES) to enhance the algorithm's exploitability.

The proposed LR-CMA-ES can be regarded as a synergy of evolutionary learning and self-learning. The evolutionary learning process consists of the evolution operators of the CMA-ES, which are detailed in Section 2.3. Self-learning is achieved through local heuristic methods. The framework of the introduced LR-CMA-ES is illustrated in Figure 5.

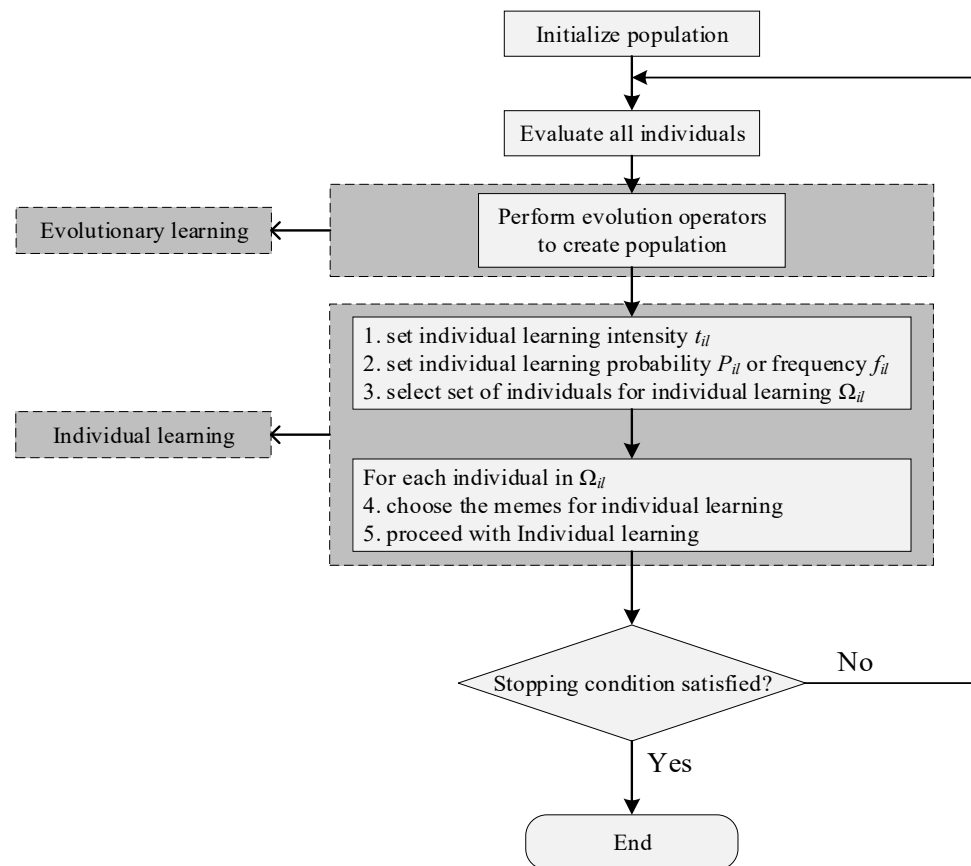


Figure 5. The framework of the introduced LR-CMA-ES.

3.3. CMA-ES with Population Reduction

To enhance scalability, the CMA-ES with population reduction (PR-CMA-ES) utilizes a relatively small population. Typically, the number of fitness evaluations is fixed. Small populations enable multiple generations and the efficient use of fitness evaluations (FEs) [28]. The optimization of multimodal problems involves two distinct tasks: locating promising attraction basins and identifying the local optimum within these basins. For the first task, the population size should be sufficiently large to account for the number of basins. Conversely, for the second task, the algorithm's solutions must be iterated through many generations to find the optimum. Therefore, the population size of the CMA-ES is defined as shown in Equation (10).

$$pop_size_t = init_{pop} * \left(\frac{[FEs - k]}{FEs} \right) \quad (10)$$

where pop_size_t is the population size of iteration t , $init_{pop}$ is the initial size of the population, FEs is the total available amount of CMA-ES, and k is the number of evaluations used in CMA-ES so far.

3.4. Pseudocode of Kbp-LaF-CMAES

The Kbp-LaF-CMAES algorithm comprises two primary procedures: the discovery of local basins and subsequent optimization within these areas. The Kbp-LaF-CMAES algorithm outlines the procedural steps involved (Algorithm 1).

In the Kbp-LaF-CMAES algorithm, a bi-population cooperative strategy is employed for exploratory searches across the problem space, complemented by the use of two CMA-ES variants for exploitation. The pseudocode illustrates the synergy between the novel global search strategy and the refined CMA-ES, facilitating a seamless transition from exploration to exploitation. This integration represents more than a mere amalgamation of strategies; it embodies an organic fusion aimed at enhancing overall algorithmic efficacy.

3.5. Complexity Analysis

The complexity of an algorithm is generally measured by the number of floating-point operations and the required iterations or recursions. In the context of metaheuristic optimization algorithms, the number of FEs is a critical aspect of complexity. The time complexity of an EA is analyzed based on a set of fitness evaluations and iterations [29]. In this paper, the complexity of Kbp-LaF-CMAES is associated with the following operations: LaF, Kbp, and two variants of CMA-ES. In the strategy of LaF, the size of two co-evolutionary populations is denoted as N_1 . The time complexity is $O(2 * N_1 * D)$. The sizes of attraction basins in each dimension need to be measured. Therefore, the complexity of LaF operation is $O(D)$. According to the time complexity of the CMA-ES, the corresponding complexity of the improved CMA-ES is $O(N_2 * D)$. N_2 is the population size of the CMA-ES algorithm. Therefore, the time complexity of Kbp-LaF-CMAES is $O(2 * N_1 * D + N_2 * D + D)$.

Algorithm 1 Kbp-LaF-CMAES

Input: the parameters configuration;
Output: optimal solution;
Initialize two populations, **Leaders** and **Followers**;
while fEval < maxEvals of Kbp-LaF **do**
 Update the population Followers;
 Merge the two populations;
 If fall into stalled **then**
 Conduct Kbp strategy;
 Update X_{best} ;
end
while not meet stop criterion **do**
 k-means clustering on Leaders;
 LR-CMA-ES for the best_k solutions;
 Update X_{best} ;
 PR-CMA-ES for the Leaders;
 Update X_{best} ;
end

3.6. Convergence Analysis of Kbp-LaF-CMAES

In the proposed algorithm, the solutions of exploration are not directly applied to the two variants of CMA-ES. Therefore, the adaptability and convergence of CMAES are not affected by the strategies of Kbp and LaF. The convergence of CMA-ES proved to be equivalent to the convergence of Kbp-LaF-CMAES.

The evolutionary process of the evolutionary algorithm can be modeled as an inhomogeneous Markovian process [30]. As a population-based stochastic optimization algorithm, the evolutionary process of CMA-ES is considered a resampling process [31]. Two primary strategies are included in CMA-ES, covariance matrix self-adaptation and step-size adaptation [32]. Then, the description of the theoretical derivation of its empirically observed convergence rate is tackled in this Section [33].

In CMA-ES, the state space $(X_t, \sigma_t) \in \mathbb{R}^n \times \mathbb{R}$ is constructed. Where $X_t \in \mathbb{R}^n$ represent the individual at iteration t . σ_t is the overall step size of an underlying sampling distribution at iteration t . Then, CMA-ES is the stochastically recursive sequence on the state space.

$$(X_{t+1}, \sigma_{t+1}) = \mathcal{F}((X_t, \sigma_t), U_t) \quad (11)$$

$\mathcal{F}$ is the transition function of the algorithm. In CMA-ES, the transition function $\mathcal{F}$ depends on the benchmark function f only through the comparison of candidate solutions. U_t is the distribution obtained by the selected solutions.

The p individuals of the population are created by a *Sol* function.

$$X_t^i = \text{Sol}((X_t, \sigma_t), U_t^i), i = 1, \dots, p \quad (12)$$

where $\text{Sol}((x, \sigma), u^i) = x + \sigma u^i$.

Then, the p candidate individuals are evaluated on f and ordered by the objective. The permutation process is denoted as $\mathcal{S} \in \mathfrak{S}(p)$. Where $\mathfrak{S}(p)$ is the permutation set of p elements.

$$\mathfrak{S}(p) \times \mathbb{U}^p \rightarrow \mathbb{U}^p \quad (13)$$

$$(\mathcal{S}, U_t) \mapsto \mathcal{S} * U_t = (U_t^{\mathfrak{S}(1)}, \dots, U_t^{\mathfrak{S}(p)}) \quad (14)$$

The update process of (X_t, σ_t) is executed according to the strategy of selection and mutation. More precisely, a measurable function $\mathcal{G}$ called the update function that maps $\mathbb{R}^n \times \mathbb{R} \times \mathbb{U}^p$ onto $\mathbb{R}^n \times \mathbb{R}$.

$$(X_{t+1}, \sigma_{t+1}) = \mathcal{G}((X_t, \sigma_t), \mathcal{S} * U_t) = \mathcal{G}((X_t, \sigma_t), Y_t) \quad (15)$$

where Y_t denotes the ordered coordinates of U_t .

CMA-ES is determined by the quadruplet $(\text{Sol}, \mathcal{G}, \mathbb{U}^p, p)$.

(1) Invariance properties of CMA-ES

Suppose that $\mathcal{M}$ is a strictly monotonic mapping, f' is the mapping function. (X'_t, σ'_t) is the Markov chain obtained when optimizing f' using the same sequence $(X_t, \sigma_t)_{t \in \mathbb{N}}$ and the same initial state $(X'_0, \sigma'_0) = (X_0, \sigma_0)$.

Proof 1. Invariance to Strictly Monotonic Transformations of f .

Assume $(X'_t, \sigma'_t) = (X_t, \sigma_t)$, and denote $X_t^i = \text{Sol}((X_t, \sigma_t), U_t^i)$. Because $\text{Ord}(f'(X_t^1), \dots, f'(X_t^p)) = \text{Ord}(f(X_t^1), \dots, f(X_t^p)) = \mathcal{S}$, the following equation is induced. Ord is the ordering function that returns the permutation set of p elements.

$$(X_{t+1}, \sigma_{t+1}) = \mathcal{G}((X_t, \sigma_t), \mathcal{S} * U_t) = \mathcal{G}((X'_t, \sigma'_t), \mathcal{S} * U_t) = (X'_{t+1}, \sigma'_{t+1}) \quad (16)$$

□

Proof 2. Invariance to Translation of f .

Φ is defined by $\Phi(x_0)(x, \sigma) = (x + x_0, \sigma)$ for all x_0, x, σ .

Then, as the homomorphism definition,

$$\mathcal{G}(\Phi(x_0)(x, \sigma), y) = \Phi(x_0)\mathcal{G}((x, \sigma), y) \quad (17)$$

$$\mathcal{G}(\Phi(x_0)(x, \sigma), \mathcal{S}_{\Phi(x_0)(x, \sigma)}^{f(x-x_0)} * u) = \Phi(x_0)\mathcal{G}((x, \sigma), \mathcal{S}_{(x, \sigma)}^{f(x)} * u) \quad (18)$$

CMA-ES algorithm is translation invariance. □

Proof 3. Invariance to Scaling of f .

φ is defined by $\varphi(\alpha)(x, \sigma) = (x/\alpha, \sigma/\alpha)$ for all x_0, x, σ .

$f(\text{Sol}((x, \sigma), u^i)) = f(\alpha \text{Sol}((\frac{x}{\alpha}, \frac{\sigma}{\alpha}), u^i))$ implies that the same permutation $\mathcal{S}$ is obtained by the (x, σ) of f and $(\frac{x}{\alpha}, \frac{\sigma}{\alpha})$ of $f(\alpha x)$, denoted as $\mathcal{S}_{(x, \sigma)}^{f(x)} = \mathcal{S}_{(\frac{x}{\alpha}, \frac{\sigma}{\alpha})}^{f(\alpha x)}$.

$$\mathcal{F}^{f(x)}((x, \sigma), u) = \mathcal{G}\left((x, \sigma), \mathcal{S}_{(x, \sigma)}^{f(x)} * u\right) \quad (19)$$

$$\mathcal{F}^{f(\alpha x)}\left(\left(\frac{x}{\alpha}, \frac{\sigma}{\alpha}\right), u\right) = \mathcal{G}\left(\left(\frac{x}{\alpha}, \frac{\sigma}{\alpha}\right), \mathcal{S}_{(\frac{x}{\alpha}, \frac{\sigma}{\alpha})}^{f(\alpha x)} * u\right) \quad (20)$$

Then,

$$\mathcal{F}^{f(x)}((x, \sigma), u) = \varphi(1/\alpha) \mathcal{F}^{f(\alpha x)}(\varphi(\alpha)(x, \sigma), u) \quad (21)$$

Hence, the property of scaling-invariance of CMA-ES is held.

(2) Linear Convergence of CMA-ES

In this part, Z_t is defined as $\frac{X_t}{\sigma_t}$. The log-progress $\ln \|X_{t+1}\| / \|X_t\|$ is investigated in the following for the analysis of linear convergence.

$$\ln \frac{\|X_{t+1}\|}{\|X_t\|} = \ln \frac{\|Z_{t+1}\| \eta^*(Y(Z_t, U_t))}{\|Z_t\|} \quad (22)$$

η^* is the multiplicative step-size update. $Y(Z_t, U_t)$ is the ordered vector $\mathcal{S}_{(Z_t, 1)} * U_t$. According to the property of logarithm, $\frac{1}{t} \ln \frac{\|X_t\|}{\|X_0\|}$ is expressed as follows.

$$\begin{aligned} \frac{1}{t} \ln \frac{\|X_t\|}{\|X_0\|} &= \frac{1}{t} \ln \prod_{k=0}^{t-1} \frac{\|X_{k+1}\|}{\|X_k\|} = \frac{1}{t} \sum_{k=0}^{t-1} \ln \frac{\|Z_{k+1}\|}{\|Z_k\|} \eta^*(Y(Z_k, U_k)) \\ &= \frac{1}{t} \sum_{k=0}^{t-1} \ln \|Z_{k+1}\| - \frac{1}{t} \sum_{k=0}^{t-1} \ln \|Z_k\| + \frac{1}{t} \sum_{k=0}^{t-1} \ln \eta^*(Y(Z_k, U_k)) \end{aligned} \quad (23)$$

$\mathcal{R}(z)$ is defined as the expectation of the logarithm of $\eta^*(Y(z, U))$. $P^t(z, A)$ is the transition probabilities of the Markov chain. The σ -finite measure π is invariant if it satisfies

$$\pi(A) = \int \pi(dz) P(z, A) \quad (24)$$

$$E_\pi \left[\ln \frac{\|X_{t+1}\|}{\|X_t\|} \right] = \int E_{U \sim p_U} [\ln(\eta^*(Y(z, U)))] \pi(dz) = \int \mathcal{R}(z) \pi(dz) \quad (25)$$

CR is the convergence rate as the opposite of RHS of the previous equation.

$$CR = - \int E_{U \sim p_U} [\ln(\eta^*(Y(z, U)))] \pi(dz) = - \int \mathcal{R}(z) \pi(dz) \quad (26)$$

Then,

$$\lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{\|X_t\|}{\|X_0\|} = \int \ln \|z\| \pi(dz) - \int \ln \|z\| \pi(dz) + \int E[\ln(\eta^*(Y(z, U)))] \pi(dz) = -CR \quad (27)$$

The linear convergence can be proven by exploiting the application of a Law of Large Numbers (LLN).

Δ is the drift operation defined as follows.

$$\Delta V(z) = \int P(z, dy) V(y) - V(z) = E_Z[V(Z_1) - V(Z_0)] \quad (28)$$

Then,

$$\|P^t(z, \cdot) - \pi\|_h \xrightarrow{t \rightarrow \infty} 0 \quad (29)$$

KbP-LaF-CMAES can reach the global optimum in probability. $\square$

4. Experiment and Analysis

The optimization capacity of the KbP-LaF-CMAES algorithm is evaluated using the CEC2013 benchmark problems. It is compared with several other metaheuristic algorithms, including LaF-CMA-ES [34], EBOwithCMAR [35], IPOP-CMAES [24], CMA-ES, and SPSR-DEMMS [36]. These test problems are categorized into three classes based on their landscape properties: unimodal, basic multimodal, and composition functions [37].

The simulation experiments were conducted on a MATLAB 2019 platform using a PC equipped with an Intel(R) Xeon(R) W-2123 CPU running at 3.6 GHz and 16 GB of memory with Windows 10 x64 OS, and the manufacturer of this device is Lenovo Group Ltd. This company is headquartered in Beijing, China. Performance was measured based on a predetermined number of fitness evaluations, which is $10,000 * D$. In Section 4.1, the typical parameters of KbP-LaF-CMAES are analyzed through a design of experiments (DOE). Section 4.2 discusses the contributions of the KbP-LaF-CMAES components based on experimental results from 28 benchmark test functions. Section 4.3 compares KbP-LaF-CMAES with five other algorithms.

Authors should interpret the results in light of previous studies and the underlying hypotheses. The findings and their implications should be discussed within the broadest possible context, and potential future research directions may also be highlighted.

4.1. Parameters Analysis

The configuration of parameters is crucial for achieving optimized results in algorithms. However, determining the optimal parameter settings can be challenging. By analyzing statistical data, the impact of parameters on performance can be assessed throughout the optimization process. In this part, the DOE method is applied to control the parameters of the proposed KbP-LaF-CMAES in the test suite of CEC2013. Three typical parameters mainly influence the capacity of the KbP-LaF-CMAES. These parameters are MainEvals (Evaluation number of LaF strategy), PertFac (Factor of perturbation), and PopFac (Factor of parent population).

A $1/2$ fractional factorial DOE was conducted for these parameters. The parameter choices are shown in Table 1: MainEvals $\in [0.3, 0.5, 0.7, 0.9]$, PertFac $\in [0.5, 1, 1.5, 2]$, and PopFac $\in [0.1, 0.25, 0.5, 0.75]$. A variety of dimensions ($D = 30$, $D = 50$, and $D = 100$) were selected as the dimensions of the test space to enhance the statistical significance. The orthogonal array of the parameter configurations is shown in Table 2. For each parameter group, the algorithm was run 10 times independently on all of the 28 functions of CEC2013.

Table 1. The choices of the parameter.

Factor Level	1	2	3	4
<i>mainEvals</i>	0.3	0.5	0.7	0.9
<i>PertFac</i>	0.5	1	1.5	2
<i>PopFac</i>	0.1	0.25	0.5	0.75

The ranks of these parameters are presented in Table 3. The MainEvals parameter is the most critical for performance. For each parameter combination in the orthogonal array, the average value (AVE) across all dimensions and the total average value (TAVE) are displayed in Table 4.

According to the experimental results, the performance variation with these parameters is shown in Figure 6. The analysis indicates that the parameter *mainEvals* is the most influential among the three parameters, with an optimal value of 0.3. The *mainEvals* parameter determines the proportion of LaF in the algorithm, which is a powerful exploration strategy in KbP-LaF-CMAES. In contrast, *PertFac* and *PopFac* are relatively minor parameters but still affect population diversity, which, in turn, influences the quality of the exploitation process. Based on Table 3 and Figure 6, the parameter values for KbP-LaF-CMAES are as follows: *mainEvals* = 0.3, *PertFac* = 1.5, and *PopFac* = 0.5.

Table 2. The combinations of the parameters.

Serial Number	<i>mainEvals</i>	<i>PertFac</i>	<i>PopFac</i>
1	0.3	0.5	0.1
2	0.3	1	0.25
3	0.3	1.5	0.50
4	0.3	2.0	0.75
5	0.5	0.5	0.25
6	0.5	1.0	0.1
7	0.5	1.5	0.75
8	0.5	2.0	0.50
9	0.7	0.5	0.5
10	0.7	1.0	0.75
11	0.7	1.5	0.1
12	0.7	2.0	0.25
13	0.9	0.5	0.75
14	0.9	1.0	0.5
15	0.9	1.5	0.25
16	0.9	2.0	0.1

Table 3. The ranks of parameters.

Level	<i>mainEvals</i>	Parameters <i>PertFac</i>	<i>PopFac</i>
1	3.2×10^3	6.5×10^3	8.1×10^3
2	6.1×10^3	7.8×10^3	6.1×10^3
3	1.0×10^4	3.3×10^3	5.5×10^3
4	8.1×10^3	8.1×10^3	7.7×10^3
Std.	2.9×10^3	2.2×10^3	1.2×10^3
Rank	1	2	3

Table 4. Performance of the parameter combination.

No.	<i>mainEvals</i>	Parameters <i>PertFac</i>	<i>PopFac</i>	30D	AVE 50D	100D	TAVE Total
1	1	1	1	6.78×10^2	1.35×10^3	7.87×10^3	3.33×10^3
2	1	2	2	4.12×10^2	8.56×10^2	6.60×10^3	2.62×10^3
3	1	3	3	2.95×10^2	7.48×10^2	1.98×10^3	1.01×10^3
4	1	4	4	4.36×10^2	8.89×10^2	1.62×10^4	5.84×10^3
5	2	1	2	6.58×10^2	1.37×10^3	2.16×10^4	7.87×10^3
6	2	2	3	3.42×10^2	8.58×10^2	1.74×10^4	6.20×10^3
7	2	3	4	4.06×10^2	8.44×10^2	6.47×10^3	2.57×10^3
8	2	4	1	5.90×10^2	1.19×10^3	2.15×10^4	7.76×10^3
9	3	1	3	4.80×10^2	1.24×10^3	1.93×10^4	7.01×10^3
10	3	2	4	3.80×10^2	3.93×10^3	3.96×10^4	1.46×10^4
11	3	3	1	4.87×10^2	1.13×10^3	1.95×10^4	7.07×10^3
12	3	4	2	6.73×10^2	2.55×10^3	3.06×10^4	1.13×10^4
13	4	1	4	6.34×10^2	1.56×10^3	2.12×10^4	7.79×10^3
14	4	2	3	6.78×10^2	1.26×10^3	2.14×10^4	7.78×10^3
15	4	3	2	3.66×10^2	7.43×10^2	6.54×10^3	2.55×10^3
16	4	4	1	7.33×10^2	4.17×10^3	3.77×10^4	1.42×10^4

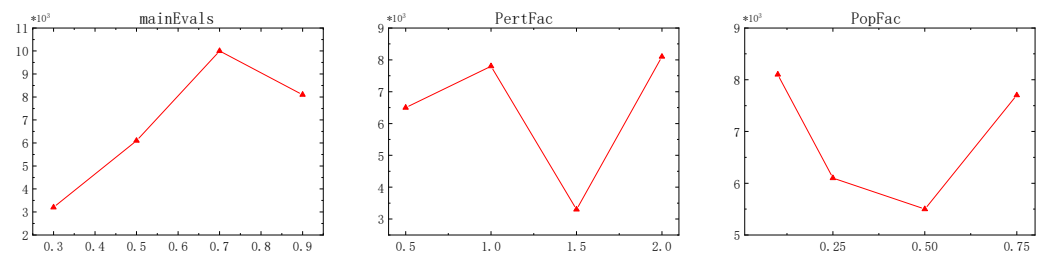


Figure 6. The tendency of the parameters.

4.2. Operator Analysis

In this Section, the effectiveness of the subcomponents of the improved method is analyzed through comparisons with CMA-ES, LaF-CMAES, and LaF. The average results of Kbp-LaF-CMAES, CMA-ES, LaF-CMAES, and LaF across three different dimensions are evaluated using the CEC2013 test suite. Figures 7–9 illustrate the performance of the four algorithms through boxplots.

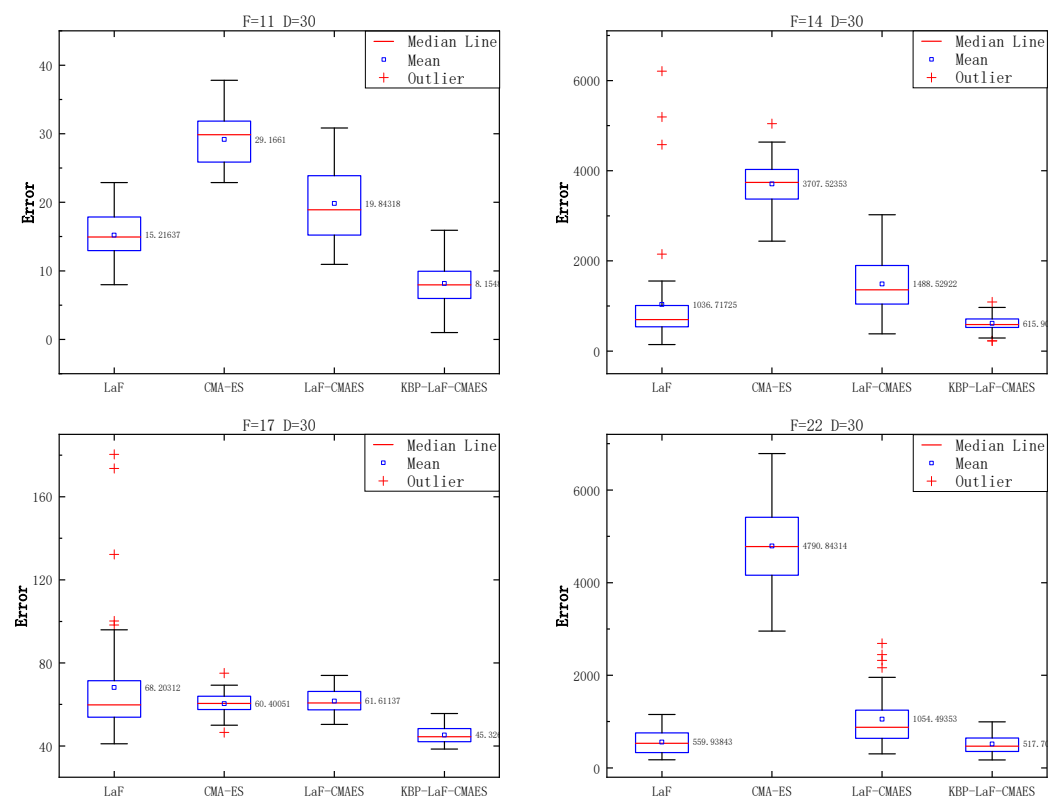


Figure 7. Boxplots of benchmark functions for each strategy (30D).

The selected test functions are Functions 11, 14, 17, and 22 from the CEC2013 benchmarks. The first three functions are multimodal problems, while Function 22 is classified as a composition function. These functions exemplify the characteristics of multimodality based on their fitness landscapes. The results indicate that all strategies contribute to the effectiveness of Kbp-LaF-CMAES.

The results indicate that the introduced method outperforms CMA-ES, LaF-CMAES, and LaF in both high-dimensional and low-dimensional functions. This finding suggests that the combination of multiple strategies yields superior effectiveness and stability compared to single-strategy algorithms.

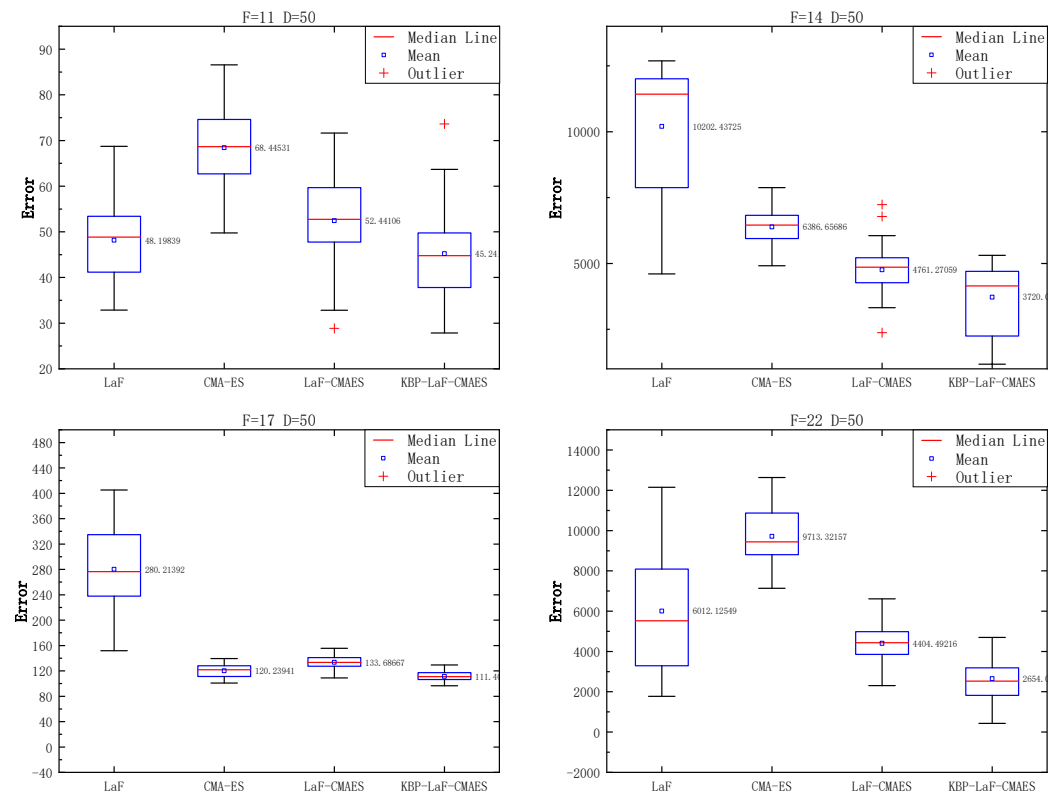


Figure 8. Boxplots of benchmark functions for each strategy (50D).

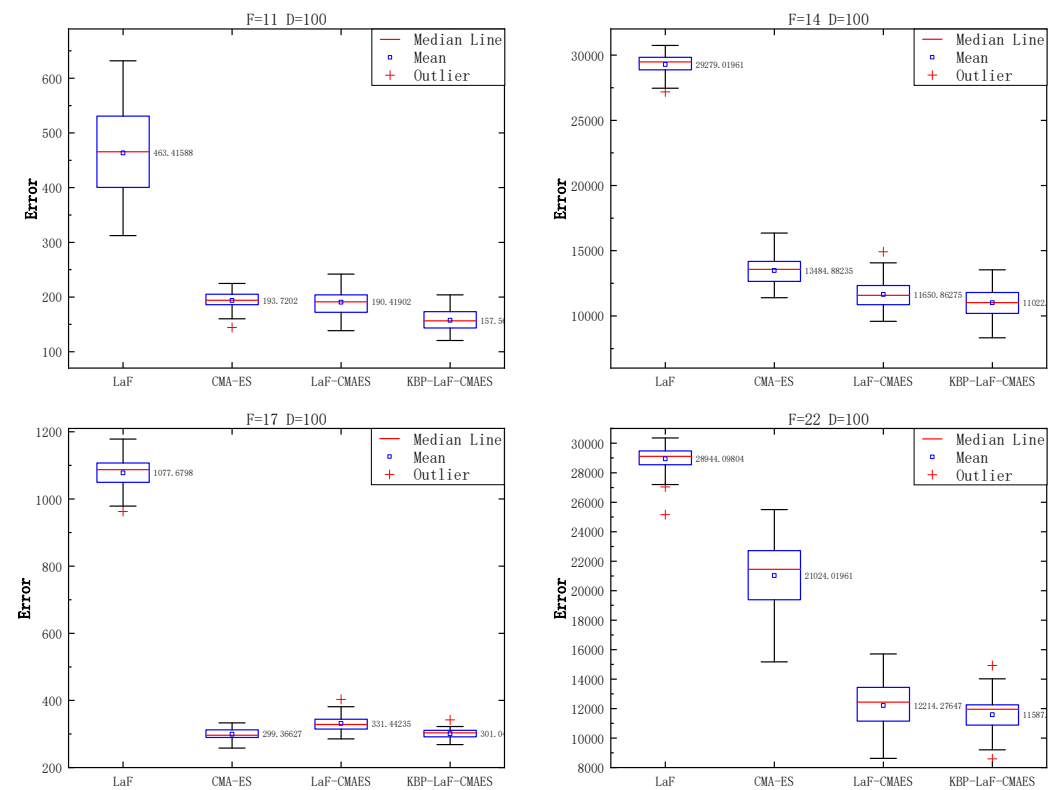


Figure 9. Boxplots of benchmark functions for each strategy (100D).

4.3. Experimental Analysis and Result

Simulations were conducted across three different dimensions. Five statistical indicators—Best, Worst, Median, Mean, and Standard Deviation (Std)—were calculated, as shown at the end of this paper. Each algorithm was run independently 51 times for each instance, and the Mean and Std indicators were derived from the performance of each algorithm on the test functions. The results of the introduced method on the CEC benchmark across the three dimensions are illustrated in Table 5, with the best performance for each function highlighted in bold. The test functions are notably complex due to their high dimensionality. Consequently, comparison experiments were performed on the test problems across these three types of dimensions to assess the robustness of the introduced algorithms.

Table 5. Friedman’s test rankings of the different algorithms.

Algorithm	Mean Rank (30D)	Mean Rank (50D)	Mean Rank (100D)
KbP-LaF-CMAES	2.75	2.48	2.41
CMA-ES	3.77	4.11	3.86
LaF-CMA-ES	3.75	3.68	3.54
IPOP-CMA-ES	3.59	3.66	3.45
EBOwithCMAR	2.77	2.77	2.70
SPSRDEMMS	4.38	4.30	5.05

The statistical variances between the introduced method and the compared algorithms are analyzed using Friedman’s test. The results of Friedman’s test for the improved method and comparison algorithms across each dimension are illustrated in Table 6. These figures consistently show KbP-LaF-CMAES achieving the highest ranking for the test benchmark. Subsequently, the significance levels of the differences between the algorithms are computed using Bonferroni–Dunn’s test.

$$CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}} \quad (30)$$

Table 6. Results of the post hoc test.

	30D		50D		100D	
	<i>p</i> -Value	Adjusted <i>p</i> -Value	<i>p</i> -Value	Adjusted <i>p</i> -Value	<i>p</i> -Value	Adjusted <i>p</i> -Value
SPSRDEMMS	0.001	0.017	0.000	0.004	0.000	0.000
CMA-ES	0.042	0.627	0.001	0.017	0.004	0.057
LaF-CMAES	0.048	0.683	0.017	0.251	0.024	0.367
IPOP-CMAES	0.093	1.000	0.018	0.276	0.038	0.575
EBOwithCMAR	0.972	1.000	0.568	1.000	0.568	1.000

In Equation (30), k and N represent the number of compared algorithms and test functions, respectively, while α denotes the confidence level, commonly set to 0.05 or 0.1. The critical difference for Bonferroni–Dunn’s test was calculated with $\alpha = 0.05$ and $\alpha = 0.1$, as shown in Equation (30).

In this study, with $k = 5$ and $N = 28$, $q_{\alpha} = 2.394$ for $\alpha = 0.05$, and $q_{\alpha} = 2.128$ for $\alpha = 0.1$. The statistical results are presented in Table 5, demonstrating that the introduced method outperforms the comparison methods. Regardless of the α value and the dimension, a significant difference is detected among the comparison methods.

The experimental results lead to several conclusions. The performance of the proposed algorithm surpasses that of most comparison algorithms across a wide range of functions in all dimensions. For low-dimensional problems (30D and 50D), the effectiveness of the introduced method is comparable to that of EBOwithCMAR. Subsequently, post hoc tests were conducted to identify the differences between these algorithms [38]. The introduced

method served as the control algorithm for these tests. The objective of the post hoc tests is to establish a comparison set of p -values. These p -values are derived from the results, with the Bonferroni–Dunn method employed to compute adjusted p -values. The test results are presented in Table 6.

Table 6 displays both the p -values and the adjusted p -values, highlighting differences among the five algorithms across all three dimensions. Specifically, in the 50D and 100D dimensions, statistical disparities have emerged between the KbP-LaF-CMAES and IPOP-CMAES algorithms. To confirm the directionality of these statistical differences, reference to the raw data is necessary.

Further statistical analysis of the simulation results involves making an all-pair comparison without a control algorithm. The outcomes of the $n \times n$ comparison using post hoc Friedman tests for the three kinds of dimensions are detailed in Tables 7–9, respectively. n is the count of the compared algorithms. From these tables, the $\frac{n \times (n-1)}{2}$ pairwise comparison results can be derived. Specifically, for the 50D and 100D test functions, the introduced method outperforms the other compared algorithms.

Table 7. Results of 6 vs. 6 comparisons on the benchmark of 30D.

	KbP-LaF-CMAES	EBOwithCMAR	IPOP-CMAES	LaF-CMAES	CMA-ES	SPSRDEMMS
KbP-LaF-CMAES		0.972	0.093	0.046	0.042	0.001
EBOwithCMAR	0.972		0.100	0.049	0.046	0.001
IPOP-CMAES	0.093	0.100		0.748	0.721	0.116
LaF-CMAES	0.046	0.049	0.748		0.972	0.211
CMA-ES	0.042	0.046	0.721	0.972		0.225
SPSRDEMMS	0.001	0.001	0.116	0.211	0.225	

Table 8. Results of 6 vs. 6 comparisons on the benchmark of 50D.

	KbP-LaF-CMAES	EBOwithCMAR	IPOP-CMAES	LaF-CMAES	CMA-ES	SPSRDEMMS
KbP-LaF-CMAES		0.391	0.018	0.017	0.001	0.000
EBOwithCMAR	0.391		0.694	0.069	0.007	0.002
IPOP-CMAES	0.018	0.694		0.972	0.372	0.199
LaF-CMAES	0.017	0.069	0.972		0.391	0.211
CMA-ES	0.001	0.007	0.372	0.391		0.694
SPSRDEMMS	0.000	0.002	0.199	0.211	0.694	

Table 9. Results of 6 vs. 6 comparisons on the benchmark of 100D.

	KbP-LaF-CMAES	EBOwithCMAR	IPOP-CMAES	LaF-CMAES	CMA-ES	SPSRDEMMS
KbP-LaF-CMAES		0.568	0.038	0.024	0.004	0.000
EBOwithCMAR	0.568		0.134	0.093	0.020	0.000
IPOP-CMAES	0.038	0.134		0.858	0.411	0.001
LaF-CMAES	0.024	0.093	0.858		0.520	0.002
CMA-ES	0.004	0.020	0.411	0.520		0.017
SPSRDEMMS	0.000	0.000	0.001	0.002	0.017	

The experimental results demonstrate a significant advantage of the introduced method over the comparison algorithms. The statistical analysis indicates that the introduced method exhibits distinct performance improvements over SPSRDEMMS and the three other variants of CMA-ES, validating its efficiency. The superiority of the introduced method becomes more evident as dimensionality increases. The comparison between KbP-LaF-CMAES and EBOwithCMAR reveals that the performance of KbP-LaF-CMAES is comparable to that of the state-of-the-art algorithm.

Detailed analysis of the experimental results shows that KbP-LaF-CMAES performs well on several typical multimodal functions, such as F7, F8, F14, F15, F18, F19, and F20.

Additionally, KbP-LaF-CMAES also excels in the two composite problems F22 and F23. For the remaining multimodal and composite problems, KbP-LaF-CMAES exhibits comparable or slightly inferior performance compared to the other algorithms. Overall, the KbP-LaF-CMAES algorithm has demonstrated preferable performance on most test functions, particularly in high-dimensional problems. The test functions exhibit strong multimodal properties, confirming that the introduced method can effectively solve multimodal problems.

Some of the comparison test results for the 30, 50, and 100 dimensions are shown in Figures 10–12, respectively. The statistical analysis validates that the introduced method performs best on the test functions across these dimensions. Although KbP-LaF-CMAES does not always find the global optimal solution, the solutions it produces are comparable to those found by other algorithms.

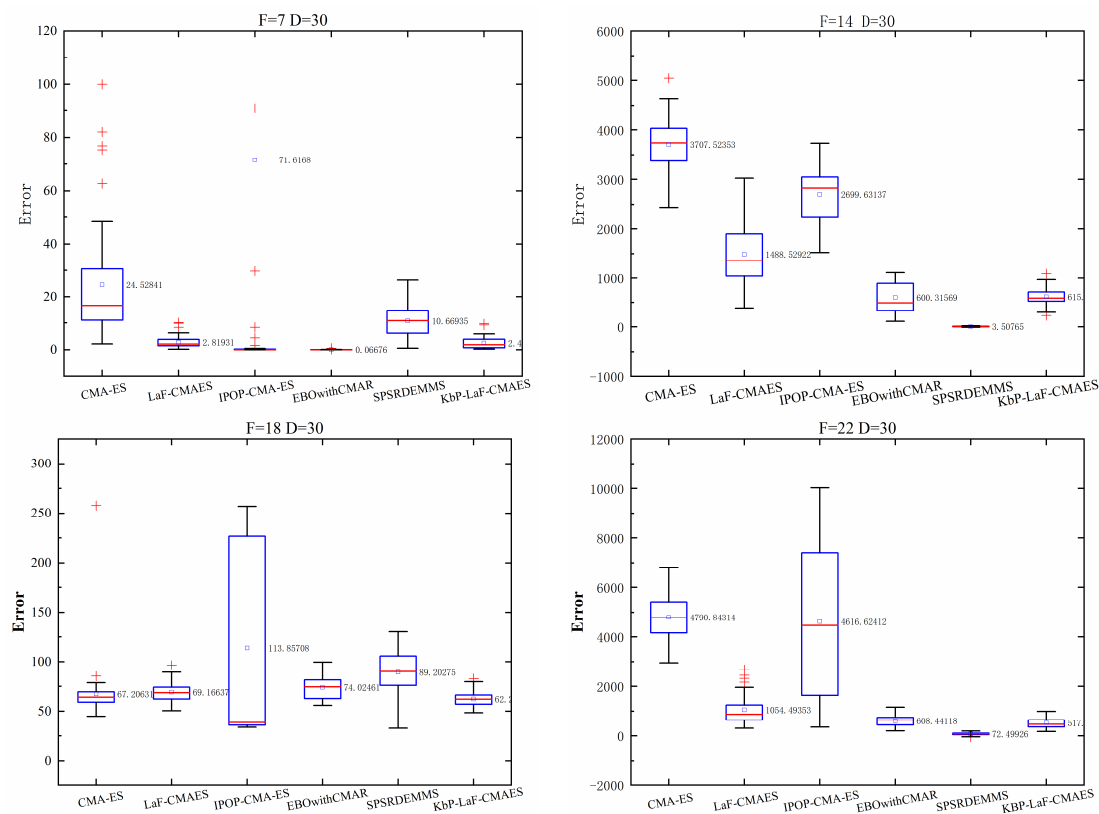


Figure 10. Boxplots of some typical benchmark functions (30D).

For multimodal functions, the introduced method achieves the best performance among the compared algorithms. On the remaining functions, KbP-LaF-CMAES maintains a comparable performance. The detailed experimental results for KbP-LaF-CMAES, CMA-ES, LaF-CMAES, IPOP-CMAES, and EBOwithCMAR are presented in Tables 10–12 for 30, 50, and 100 dimensions, respectively, where the bolded data in the table indicates the optimal results.

In summary, the introduced method has better Friedman's test results on all three dimensions of the CEC benchmark relative to the other five compared algorithms, which proves that the robustness of the introduced algorithm is the best relative to the other five compared algorithms, and the introduced method outperforms several comparison algorithms, including SPSRDEMMS and three variants of CMA-ES, especially as the dimensionality increases. It performs well on many multimodal and composite functions, and its overall performance is preferable in high-dimensional problems. Although it does not always find the global optimal solution, its results are comparable to other leading algorithms, particularly excelling in multimodal functions.

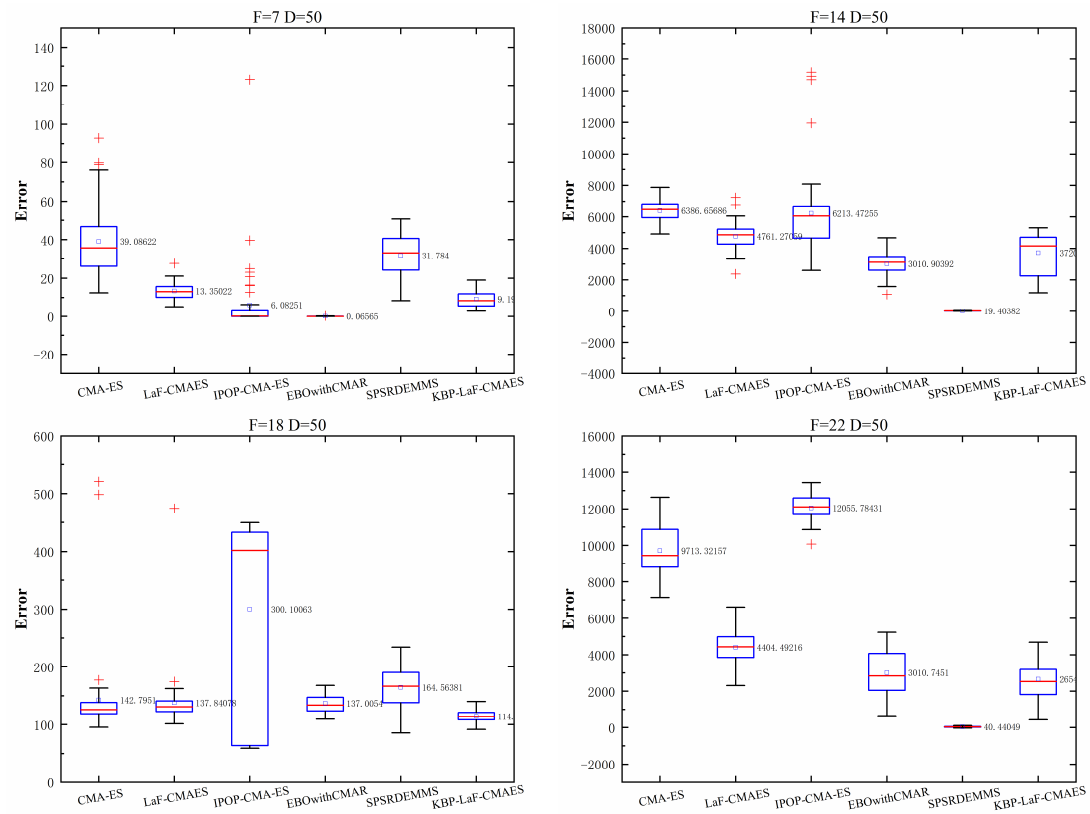


Figure 11. Boxplots of some typical benchmark functions (50D).

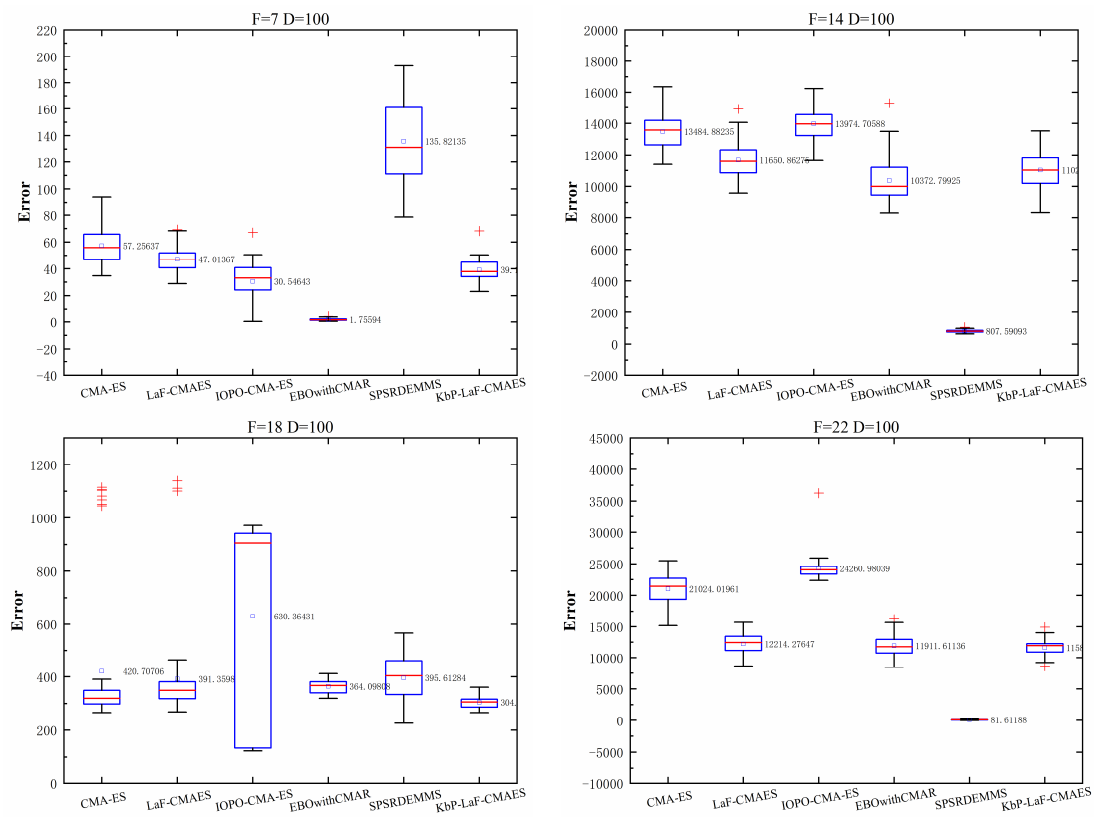


Figure 12. Boxplots of some typical benchmark functions (100D).

Table 10. The results of different algorithms for 30-dimensional benchmark functions.

Function	Criterion	CMA-ES	LaF-CMA-ES	IPOP-CMAES	EBOwithCMAR	SPSRDEMMS	KbP-LaF-CMAES
1	Mean	0	0	0	0	0	0
	Std. Dev.	0	0	0	0	0	0
2	Mean	0	0	0	2.28×10^{-4}	1.02×10^5	0
	Std. Dev.	0	0	0	3.47×10^{-4}	5.30×10^4	0
3	Mean	4.39×10^1	2.80×10^1	1.22×10^2	3.78×10^{-7}	1.10×10^7	1.29×10^{-1}
	Std. Dev.	2.92×10^2	1.50×10^2	7.42×10^2	2.40×10^{-6}	1.39×10^7	6.21×10^{-1}
4	Mean	0	0	0	5.46×10^{-5}	2.41 3.28	0
	Std. Dev.	0	0	0	1.17×10^{-4}		0
5	Mean	0	0	0	0	0	0
	Std. Dev.	0	0	0	0	0	0
6	Mean	0	1.11	3.08×10^{-1}	0	1.74×10^1	1.03
	Std. Dev.	0	5.19	2.20	0	1.14×10^1	5.16
7	Mean	2.45×10^1	2.82	7.16×10^1	6.66×10^{-2}	1.10×10^1	2.47
	Std. Dev.	2.18×10^1	2.21	4.91×10^2	1.31×10^{-1}	6.18	2.16
8	Mean	2.12×10^1	2.10×10^1	2.12×10^1	2.09×10^1	2.10×10^1	2.10×10^1
	Std. Dev.	5.58×10^{-2}	4.39×10^{-2}	6.65×10^{-2}	1.03	5.01×10^{-2}	5.97×10^{-2}
9	Mean	2.14×10^1	1.10×10^1	6.77	2.39×10^1	2.49×10^1	1.18×10^1
	Std. Dev.	3.10	2.52	2.08	1.68	3.45	2.91
10	Mean	8.70×10^{-4}	5.12×10^{-3}	0	1.45×10^{-4}	5.40×10^{-2}	7.25×10^{-4}
	Std. Dev.	2.41×10^{-3}	3.72×10^{-3}	0	1.04×10^{-3}	4.02×10^{-2}	2.22×10^{-3}
11	Mean	2.92×10^1	1.98×10^1	6.05	1.00×10^1	0	8.15
	Std. Dev.	3.89	4.98	1.72	0	0	3.02
12	Mean	2.87×10^1	2.66×10^1	5.64	3.37×10^1	4.27×10^1	2.65×10^1
	Std. Dev.	4.04	5.09	2.01	2.71	1.36×10^1	6.01
13	Mean	7.14×10^1	5.90×10^1	7.04	6.80×10^1	7.98×10^1	5.22×10^1
	Std. Dev.	1.29×10^1	1.93×10^1	4.68	1.17×10^1	2.09×10^1	1.53×10^1
14	Mean	3.71×10^3	1.49×10^3	2.70×10^3	6.00×10^2	3.26	6.16×10^2
	Std. Dev.	5.11×10^2	6.81×10^2	5.13×10^2	4.56×10^2	6.13	1.92×10^2
15	Mean	3.71×10^3	2.99×10^3	2.65×10^3	2.58×10^3	4.42×10^3	2.46×10^3
	Std. Dev.	4.73	5.08	4.61	3.62×10^2	6.98×10^2	4.38×10^2
16	Mean	3.72×10^{-2}	5.40×10^{-2}	1.50	1.04×10^{-1}	2.28	5.60×10^{-2}
	Std. Dev.	1.53×10^{-2}	2.73×10^{-2}	2.56	7.35×10^{-2}	3.74×10^{-1}	3.40×10^{-2}
17	Mean	6.04×10^1	6.16×10^1	3.95×10^1	3.04×10^1	3.04×10^1	4.53×10^1
	Std. Dev.	5.20	5.97	2.82×10^1	2.44×10^1	7.58×10^{-3}	4.47
18	Mean	6.72×10^1	6.92×10^1	1.14×10^2	7.40×10^1	8.93×10^1	6.22×10^1
	Std. Dev.	2.83×10^1	9.31	9.57×10^1	1.26×10^1	2.07×10^1	7.10
19	Mean	2.26	2.52	2.85	1.12	1.16	2.09
	Std. Dev.	3.69×10^{-1}	4.67×10^{-1}	4.26×10^{-1}	1.57×10^{-1}	1.79×10^{-1}	3.32×10^{-1}
20	Mean	1.49×10^1	1.14×10^1	1.46×10^1	1.03×10^1	1.12×10^1	1.00×10^1
	Std. Dev.	1.65×10^{-1}	6.50×10^{-1}	9.82×10^{-1}	1.32	5.25×10^{-1}	6.90×10^{-1}
21	Mean	1.88×10^2	3.04×10^2	2.82×10^2	2.99×10^2	2.85×10^2	2.82×10^2
	Std. Dev.	3.25×10^1	8.56×10^1	3.83×10^1	5.13×10^1	6.94×10^1	6.57×10^1
22	Mean	4.79×10^3	1.05×10^3	4.62×10^3	6.08×10^2	7.66×10^1	5.18×10^2
	Std. Dev.	8.25×10^2	5.54×10^2	3.06×10^3	2.29×10^2	4.88×10^1	1.96×10^2
23	Mean	5.16×10^3	3.00×10^3	5.24×10^3	2.19×10^3	4.77×10^3	2.59×10^3
	Std. Dev.	6.31×10^2	6.55×10^2	1.33×10^3	4.10×10^2	7.75×10^2	4.86×10^2
24	Mean	1.92×10^2	2.10×10^2	2.00×10^2	1.97×10^2	2.53×10^2	2.08×10^2
	Std. Dev.	4.01×10^1	7.28	0	1.08×10^1	8.98	7.63
25	Mean	2.67×10^2	2.54×10^2	2.45×10^2	2.40×10^2	2.64×10^2	2.50×10^2
	Std. Dev.	6.95	6.97	8.63	3.92	8.34	1.16×10^1
26	Mean	1.76×10^2	2.00×10^2	3.00×10^2	2.00×10^2	2.00×10^2	2.00×10^2
	Std. Dev.	4.03×10^1	0	1.72	1.20×10^{-5}	4.94×10^{-3}	0
27	Mean	6.19×10^2	4.03×10^2	3.01×10^2	3.00×10^2	8.88×10^2	4.07×10^2
	Std. Dev.	1.20×10^2	6.37×10^1	3.47	3.51×10^{-1}	9.20×10^1	8.14×10^1
28	Mean	2.22×10^2	3.00×10^2	3.00×10^2	2.73×10^2	3.00×10^2	3.00×10^2
	Std. Dev.	9.86×10^1	0	0	6.95×10^1	3.94×10^{-13}	0

Table 11. The results of different algorithms for 50-dimensional benchmark functions.

Function	Criterion	CMA-ES	LaF-CMA-ES	IPOP-CMAES	EBOwithCMAR	SPSRDEMMS	KbP-LaF-CMAES
1	Mean	0	0	0	0	0	0
	Std. Dev.	0	0	0	0	0	0
2	Mean	0	0	3.38×10^2	2.20×10^{-3}	5.65×10^5	0
	Std. Dev.	0	0	2.67×10^2	2.57×10^{-3}	2.56×10^5	0
3	Mean	1.45×10^3	9.80×10^3	3.11×10^5	1.16×10^3	4.44×10^7	2.68
	Std. Dev.	5.65×10^3	3.94×10^4	2.21×10^6	2.45×10^3	4.56×10^7	1.54×10^1
4	Mean	0	0	0	1.97×10^{-4}	5.17	0
	Std. Dev.	0	0	0	3.69×10^{-4}	4.72	0
5	Mean	0	0	1.21	0	0	0
	Std. Dev.	0	0	8.62	0	0	0
6	Mean	3.54×10^1	4.34×10^1	4.34×10^1	4.34×10^1	4.37×10^1	4.34×10^1
	Std. Dev.	1.66×10^1	0	0	4.81×10^{-14}	1.11	0
7	Mean	3.91×10^1	1.34×10^1	6.08	6.57×10^{-2}	3.17×10^1	9.19
	Std. Dev.	1.81×10^1	4.50	1.86×10^1	7.14×10^1	9.20	4.33
8	Mean	2.14×10^1	2.12×10^1	2.14×10^1	2.11×10^1	2.11×10^1	2.12×10^1
	Std. Dev.	4.56×10^{-2}	5.09×10^{-2}	5.21×10^{-2}	7.59×10^{-2}	4.19×10^{-2}	4.07×10^{-2}
9	Mean	4.09×10^1	2.37×10^1	1.32×10^1	4.86×10^1	5.12×10^1	2.15×10^1
	Std. Dev.	3.43	3.50	2.68	3.55	4.97	4.52
10	Mean	7.25×10^{-4}	3.38×10^{-3}	0	6.14×10^{-3}	5.86×10^{-2}	1.45×10^{-4}
	Std. Dev.	2.22×10^{-3}	4.24×10^{-3}	0	6.67×10^{-3}	3.89×10^{-2}	1.04×10^{-3}
11	Mean	6.85×10^1	5.24×10^1	1.33×10^1	5.00×10^1	0	4.52×10^1
	Std. Dev.	8.76	9.48	3.10	0	0	9.79
12	Mean	6.76×10^1	6.68×10^1	1.21×10^1	6.49×10^1	8.74×10^1	5.43×10^1
	Std. Dev.	8.64	8.97	2.02	5.87	2.16×10^1	9.95
13	Mean	1.61×10^2	1.61×10^2	2.41×10^1	1.39×10^2	1.59×10^2	1.28×10^2
	Std. Dev.	2.66×10^1	2.88×10^1	9.54	1.49×10^1	2.60×10^1	1.94×10^1
14	Mean	6.39×10^3	4.76×10^3	6.21×10^3	3.01×10^3	2.07×10^1	3.72×10^3
	Std. Dev.	6.72×10^2	8.66×10^2	2.70×10^3	8.72×10^2	1.35×10^1	1.21×10^3
15	Mean	7.11×10^3	5.98×10^3	8.23×10^3	6.09×10^3	8.63×10^3	5.63
	Std. Dev.	7.75×10^2	8.44×10^2	1.87×10^3	8.97×10^2	8.47×10^2	6.17
16	Mean	3.50×10^{-2}	4.40×10^{-2}	1.14	5.66×10^{-2}	2.83	2.86×10^{-2}
	Std. Dev.	1.08×10^{-2}	1.92×10^{-2}	2.41	2.61×10^{-2}	6.01×10^{-1}	1.33×10^{-2}
17	Mean	1.20×10^2	1.34×10^2	8.51×10^1	5.16×10^1	5.08×10^1	1.11×10^2
	Std. Dev.	9.68	9.43	8.90×10^1	5.56×10^{-1}	2.44×10^{-2}	7.37
18	Mean	1.43×10^2	1.38×10^2	3.00×10^2	1.37×10^2	1.57×10^2	1.14×10^2
	Std. Dev.	7.66×10^1	5.01×10^1	1.70×10^2	1.55×10^1	3.65×10^1	1.10×10^1
19	Mean	4.69	5.03	5.08	2.48	1.96	4.23
	Std. Dev.	6.32×10^{-1}	1.04	8.84×10^{-1}	3.80×10^{-1}	3.13×10^{-1}	5.77×10^{-1}
20	Mean	2.47×10^1	2.10×10^1	2.26×10^1	1.98×10^1	2.06×10^1	1.98×10^1
	Std. Dev.	3.37×10^{-1}	7.58×10^{-1}	2.26	8.57×10^{-1}	7.92×10^{-1}	7.17×10^{-1}
21	Mean	1.98×10^2	6.14×10^2	3.87×10^2	6.32×10^2	6.06×10^2	5.82×10^2
	Std. Dev.	1.40×10^1	4.22×10^2	3.15×10^2	3.64×10^2	4.42×10^2	4.34×10^2
22	Mean	9.71×10^3	4.40×10^3	1.21×10^4	3.01×10^3	3.94×10^1	2.65×10^3
	Std. Dev.	1.30×10^3	9.40×10^2	6.87×10^2	1.53×10^3	2.97×10^1	1.03×10^3
23	Mean	9.56×10^3	6.45×10^3	1.13×10^4	5.04×10^3	8.91×10^3	5.79×10^3
	Std. Dev.	1.37×10^3	1.08×10^3	5.40×10^2	7.19×10^2	9.56×10^2	5.83×10^2
24	Mean	2.71×10^2	2.28×10^2	2.00×10^2	2.00×10^2	3.11×10^2	2.30×10^2
	Std. Dev.	1.63×10^1	1.07×10^1	3.89×10^{-1}	5.11×10^{-2}	1.43×10^1	1.09×10^1
25	Mean	3.40×10^2	2.93×10^2	2.83×10^2	2.71×10^2	3.35×10^2	2.90×10^2
	Std. Dev.	1.15×10^1	9.15	7.87	5.27	1.29×10^1	9.93
26	Mean	2.07×10^2	2.59×10^2	2.99×10^2	2.00×10^2	2.87×10^2	2.50×10^2
	Std. Dev.	3.92×10^1	6.93×10^1	1.48×10^1	2.65×10^{-4}	1.10×10^2	6.58×10^1
27	Mean	1.08×10^3	7.46×10^2	4.58×10^2	3.06×10^2	1.53×10^3	7.14×10^2
	Std. Dev.	9.85×10^1	1.54×10^2	1.93×10^2	7.87	1.45×10^2	1.37×10^2
28	Mean	4.53×10^2	4.00×10^2	4.01×10^2	4.00×10^2	7.54×10^2	4.00×10^2
	Std. Dev.	4.22×10^2	0	5.36	2.87×10^{-13}	9.78×10^2	0

Table 12. The results of different algorithms for 100-dimensional benchmark functions.

Function	Criterion	CMA-ES	LaF-CMA-ES	IPOP-CMAES	EBOwithCMAR	SPSRDEMMS	KbP-LaF-CMAES
1	Mean	0	0	0	0	0	0
	Std. Dev.	0	0	0	0	0	0
2	Mean	0	1.00×10^{-6}	2.31×10^6	4.96×10^{-3}	7.13×10^5	0
	Std. Dev.	0	2.00×10^{-6}	8.03×10^5	1.80×10^{-3}	3.42×10^5	0
3	Mean	1.69×10^5	8.74×10^5	1.63×10^5	6.93×10^5	5.12×10^7	1.35×10^4
	Std. Dev.	3.07×10^5	9.47×10^5	2.39×10^5	1.10×10^6	4.91×10^7	2.79×10^4
4	Mean	0	0	0	4.56×10^{-3}	7.14	0
	Std. Dev.	0	0	0	9.26×10^{-3}	5.26	0
5	Mean	0	1.00×10^{-6}	1.40×10^{-4}	0	0	0
	Std. Dev.	0	0	7.6×10^{-5}	0	0	0
6	Mean	4.13	7.43×10^1	2.06×10^2	1.58×10^2	4.71×10^1	5.84×10^1
	Std. Dev.	1.73×10^1	8.61×10^1	5.45×10^1	9.09×10^1	8.15×10^1	7.77×10^1
7	Mean	5.73×10^1	4.70×10^1	3.06×10^1	1.76	1.37×10^2	3.92×10^1
	Std. Dev.	1.32×10^1	8.60	1.46×10^1	7.80×10^{-1}	3.22×10^1	7.79
8	Mean	2.14×10^1	2.13×10^1	2.15×10^1	2.13×10^1	2.13×10^1	2.13×10^1
	Std. Dev.	4.29×10^{-2}	2.74×10^{-2}	4.92×10^{-2}	3.75×10^{-2}	4.19×10^{-2}	2.47×10^{-2}
9	Mean	9.15×10^1	8.14×10^1	3.01×10^1	1.28×10^2	1.52×10^2	7.68×10^1
	Std. Dev.	7.70	7.48	3.91	4.21	7.51	6.43
10	Mean	4.35×10^{-4}	3.14×10^{-3}	2.90×10^{-3}	1.23×10^{-2}	1.32×10^{-1}	2.90×10^{-4}
	Std. Dev.	1.76×10^{-3}	3.82×10^{-3}	4.03×10^{-3}	8.35×10^{-3}	6.91×10^{-2}	1.45×10^{-3}
11	Mean	1.94×10^2	1.90×10^2	3.48×10^1	1.77×10^2	0	1.58×10^2
	Std. Dev.	1.53×10^1	2.22×10^1	5.05	7.67	0	2.00×10^1
12	Mean	1.83×10^2	2.31×10^2	3.33×10^1	3.13×10^2	2.69×10^2	1.99×10^2
	Std. Dev.	1.43×10^1	2.50×10^1	3.64	5.65	3.63×10^1	2.23×10^1
13	Mean	3.99×10^2	4.74×10^2	6.91×10^1	9.80×10^2	4.72×10^2	3.98×10^2
	Std. Dev.	4.05×10^1	3.74×10^1	1.99×10^1	3.07×10^2	5.91×10^1	3.91×10^1
14	Mean	1.35×10^4	1.17×10^4	1.40×10^4	1.04×10^4	7.93×10^2	1.10×10^4
	Std. Dev.	1.03×10^3	1.21×10^3	9.62×10^2	1.37×10^3	8.32×10^1	1.12×10^3
15	Mean	1.27×10^4	1.13×10^4	1.20×10^4	1.36×10^4	1.82×10^4	1.11×10^4
	Std. Dev.	1.27×10^3	1.13×10^3	1.39×10^3	1.29×10^3	1.79×10^3	1.10×10^3
16	Mean	3.24×10^{-2}	3.93×10^{-2}	8.59×10^{-1}	3.70×10^{-2}	2.19	2.68×10^{-2}
	Std. Dev.	8.70×10^{-3}	1.30×10^{-2}	2.17	1.67×10^{-2}	8.11×10^{-1}	5.74×10^{-3}
17	Mean	2.99×10^2	3.31×10^2	1.33×10^2	1.13×10^2	1.26×10^2	3.01×10^2
	Std. Dev.	1.77×10^1	2.52×10^1	4.19	1.10×10^1	5.31×10^{-1}	1.43×10^1
18	Mean	4.21×10^2	3.91×10^2	6.30×10^2	3.64×10^2	3.98×10^2	3.04×10^2
	Std. Dev.	2.68×10^2	1.87×10^2	3.85×10^2	2.54×10^1	7.72×10^1	2.18×10^1
19	Mean	1.19×10^1	1.37×10^1	1.35×10^1	7.52	5.34×10^1	1.20×10^1
	Std. Dev.	1.33	2.20	1.75×10^1	8.63×10^{-1}	7.35×10^{-1}	1.30
20	Mean	5.00×10^1	5.00×10^1	5.00×10^1	5.00×10^1	5.87×10^1	5.00×10^1
	Std. Dev.	0	0	0	1.10×10^{-12}	0	0
21	Mean	2.88×10^2	3.61×10^2	3.88×10^2	3.90×10^2	3.88×10^2	3.67×10^2
	Std. Dev.	3.25×10^1	4.93×10^1	3.25×10^1	3.00×10^1	4.98×10^1	4.76×10^1
22	Mean	2.10×10^4	1.22×10^4	2.43×10^4	1.19×10^4	8.31×10^1	1.16×10^4
	Std. Dev.	2.37×10^3	1.52×10^3	1.90×10^3	1.80×10^3	4.09×10^1	1.27×10^3
23	Mean	1.95×10^3	1.31×10^4	2.08×10^4	1.22×10^4	1.72×10^4	1.30×10^4
	Std. Dev.	1.82×10^3	1.63×10^3	8.95×10^2	1.61×10^3	1.95×10^3	1.24×10^3
24	Mean	3.80×10^2	3.01×10^2	2.00×10^2	2.02×10^2	3.58×10^2	2.89×10^2
	Std. Dev.	2.09×10^1	1.21×10^1	5.61×10^{-2}	8.25×10^{-1}	1.92×10^1	1.11×10^1
25	Mean	5.00×10^2	4.22×10^2	3.71×10^2	3.59×10^2	5.73×10^2	4.10×10^2
	Std. Dev.	1.70×10^1	1.51×10^1	1.11×10^1	1.00×10^1	2.78×10^1	1.20×10^1
26	Mean	3.46×10^2	4.34×10^2	3.13×10^2	3.08×10^2	5.43×10^2	4.19×10^2
	Std. Dev.	1.33×10^2	1.49×10^1	1.50×10^1	3.86	5.01×10^1	1.41×10^1
27	Mean	2.11×10^3	1.56×10^3	4.33×10^2	1.53×10^3	2.71×10^3	1.41×10^3
	Std. Dev.	1.48×10^2	1.96×10^2	2.44×10^2	1.93×10^2	4.12×10^2	1.54×10^2
28	Mean	2.83×10^3	2.97×10^3	3.08×10^3	2.46×10^3	5.72×10^3	3.03×10^3
	Std. Dev.	6.72×10^2	8.47×10^2	9.48×10^2	2.33×10^1	1.62×10^3	8.98×10^2

5. Conclusions

In this paper, KbP, LaF strategies, and two variants of CMA-ES are integrated into the introduced method to solve MMOPs. These strategies are employed to balance exploration and exploitation. Simulation results on the CEC problems demonstrate that the performance of the introduced method surpasses that of five other compared algorithms. Statistical analysis indicates that the introduced method significantly differs from the other four compared algorithms. Its performance is comparable to that of EBOwithCMAR. Com-

parative analysis confirms that the introduced method is an efficient and effective algorithm for MMOPs.

In multimodal problems, the complexity of the landscape increases the difficulty of the problems. The strategy of multi-objective evolutionary algorithms is a promising research direction for solving multimodal problems. Additionally, landscape analysis methods can be integrated with CMA-ES to measure the size of attraction basins. For the application of CMA-ES, its framework can be extended to combinatorial optimization problems, such as distributed production scheduling.

Author Contributions: Conceptualization, H.L.; Methodology, H.L.; Software, H.L.; Validation, L.Q. and Z.Z.; Investigation, L.Q.; Writing—original draft, H.L.; Writing—review & editing, L.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Gansu Natural Science Foundation grant number [21JR7RA204, 1506RJZA007] and Gansu Province Higher Education Innovation Foundation [2022B-107, 2019A-056].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Luo, W.; Qiao, Y.; Lin, X.; Xu, P.; Preuss, M. Hybridizing Niching, Particle Swarm Optimization, and Evolution Strategy for Multimodal Optimization. *IEEE Trans. Cybern.* **2022**, *52*, 6707–6720. [[CrossRef](#)] [[PubMed](#)]
2. Tanabe, R.; Ishibuchi, H. A Review of Evolutionary Multimodal Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2020**, *24*, 193–200. [[CrossRef](#)]
3. Li, X.; Ren, J. MICQ-IPSO: An effective two-stage hybrid feature selection algorithm for high-dimensional data. *Neurocomputing* **2022**, *501*, 328–342. [[CrossRef](#)]
4. Zhao, F.; He, X.; Wang, L. A Two-Stage Cooperative Evolutionary Algorithm With Problem-Specific Knowledge for Energy-Efficient Scheduling of No-Wait Flow-Shop Problem. *IEEE Trans. Cybern.* **2021**, *51*, 5291–5303. [[CrossRef](#)]
5. Zhao, F.; Ma, R.; Wang, L. A Self-Learning Discrete Jaya Algorithm for Multiobjective Energy-Efficient Distributed No-Idle Flow-Shop Scheduling Problem in Heterogeneous Factory System. *IEEE Trans. Cybern.* **2021**, *52*, 12675–12686. [[CrossRef](#)]
6. Wang, Z.J.; Zhan, Z.H.; Lin, Y.; Yu, W.J.; Wang, H.; Kwong, S.; Zhang, J. Automatic Niching Differential Evolution With Contour Prediction Approach for Multimodal Optimization Problems. *IEEE Trans. Evol. Comput.* **2020**, *24*, 114–128. [[CrossRef](#)]
7. Hu, Z.; Zhou, T.; Su, Q.; Liu, M. A niching backtracking search algorithm with adaptive local search for multimodal multiobjective optimization. *Swarm Evol. Comput.* **2022**, *69*, 101031. [[CrossRef](#)]
8. Yazdinejad, A.; Dehghantanha, A.; Parizi, R.M.; Epiphaniou, G. An optimized fuzzy deep learning model for data classification based on NSGA-II. *Neurocomputing* **2023**, *522*, 116–128. [[CrossRef](#)]
9. Peng, Y.; Ishibuchi, H. A Diversity-Enhanced Subset Selection Framework for Multimodal Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2022**, *26*, 886–900. [[CrossRef](#)]
10. Liu, M.; Liu, J.; Hu, Z.; Ge, Y.; Nie, X. Bid optimization using maximum entropy reinforcement learning. *Neurocomputing* **2022**, *501*, 529–543. [[CrossRef](#)]
11. Su, H.; Zhao, D.; Heidari, A.A.; Liu, L.; Zhang, X.; Mafarja, M.; Chen, H. RIME: A physics-based optimization. *Neurocomputing* **2023**, *532*, 183–214. [[CrossRef](#)]
12. Zhao, F.; Di, S.; Cao, J.; Tang, J.; Jonrinaldi. A novel cooperative multi-stage hyper-heuristic for combination optimization problems. *Complex Syst. Model. Simul.* **2021**, *1*, 91–108. [[CrossRef](#)]
13. Chen, Z.; Zhan, Z.; Wang, H.; Zhang, J. Distributed Individuals for Multiple Peaks: A Novel Differential Evolution for Multimodal Optimization Problems. *IEEE Trans. Evol. Comput.* **2020**, *24*, 708–719. [[CrossRef](#)]
14. Wang, Z.-J.; Zhou, Y.-R.; Zhang, J. Adaptive Estimation Distribution Distributed Differential Evolution for Multimodal Optimization Problems. *IEEE Trans. Cybern.* **2022**, *52*, 6059–6070. [[CrossRef](#)] [[PubMed](#)]
15. Agrawal, S.; Tiwari, A. Solving multimodal optimization problems using adaptive differential evolution with archive. *Inf. Sci.* **2022**, *612*, 1024–1044. [[CrossRef](#)]
16. Sheng, M.; Chen, S.; Liu, W.; Mao, J.; Liu, X. A differential evolution with adaptive neighborhood mutation and local search for multi-modal optimization. *Neurocomputing* **2022**, *489*, 309–322. [[CrossRef](#)]
17. Lin, X.; Luo, W.; Xu, P. Differential Evolution for Multimodal Optimization with Species by Nearest-Better Clustering. *IEEE Trans. Cybern.* **2021**, *51*, 970–983. [[CrossRef](#)] [[PubMed](#)]
18. Ahrari, A.; Deb, K.; Preuss, M. Multimodal Optimization by Covariance Matrix Self-Adaptation Evolution Strategy with Repelling Subpopulations. *Evol. Comput.* **2017**, *25*, 439–471. [[CrossRef](#)] [[PubMed](#)]

19. Li, W.; Xu, Q. Covariance Matrix adaptation based on Opposition learning for multimodal optimization. In Proceedings of the Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 23–26. [\[CrossRef\]](#)
20. Liu, H.; Zhang, J. Multimodal LaF-CMA-ES algorithm based on homotopic convex transformation. In Proceedings of the 5th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM 2023), Brussels, Belgium, 20–21 October 2023.
21. Zhao, F.; Qin, S.; Yang, G.; Ma, W.; Zhang, C.; Song, H. A factorial based particle swarm optimization with a population adaptation mechanism for the no-wait flow shop scheduling problem with the makespan objective. *Expert Syst. Appl.* **2019**, *126*, 41–53. [\[CrossRef\]](#)
22. Zhao, F.; Xue, F.; Zhang, Y.; Ma, W.; Zhang, C.; Song, H. A hybrid algorithm based on self-adaptive gravitational search algorithm and differential evolution. *Expert Syst. Appl.* **2018**, *113*, 515–530. [\[CrossRef\]](#)
23. Gonzalez-Fernandez, Y.; Chen, S. Leaders and followers—A new metaheuristic to avoid the bias of accumulated information. In Proceedings of the IEEE Congress on Evolutionary Computation, Sendai, Japan, 25–28 May 2015; pp. 776–783. [\[CrossRef\]](#)
24. Auger, A.; Hansen, N. A restart CMA evolution strategy with increasing population size. In Proceedings of the IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2–5 September 2005; pp. 1769–1776. [\[CrossRef\]](#)
25. Chen, S.; Abdulsalam, I.; Yadollahpour, N.; Gonzalez Fernandez, Y. Particle Swarm optimization with pbest Perturbations. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8. [\[CrossRef\]](#)
26. Hansen, N.; Ostermeier, A. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 20–22 May 1996; pp. 312–317. [\[CrossRef\]](#)
27. Quang Huy, N.; Yew-Soon, O.; Meng Hiot, L. A Probabilistic Memetic Framework. *IEEE Trans. Evol. Comput.* **2009**, *13*, 604–623. [\[CrossRef\]](#)
28. Bolufe Rohler, A.; Fiol Gonzalez, S.; Chen, S. A minimum population search hybrid for large scale global optimization. In Proceedings of the IEEE Congress on Evolutionary Computation, Sendai, Japan, 25–28 May 2015; pp. 1958–1965. [\[CrossRef\]](#)
29. Ren, Z.; Zhang, A.; Wen, C.; Feng, Z. A Scatter Learning Particle Swarm Optimization Algorithm for Multimodal Problems. *IEEE Trans. Cybern.* **2014**, *44*, 1127–1140. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Beyer, H.-G. *The Theory of Evolution Strategies*; Springer: Berlin/Heidelberg, Germany, 2001; Volume 39.
31. Zhao, F.; Zhao, L.; Wang, L.; Song, H. An ensemble discrete differential evolution for the distributed blocking flowshop scheduling with minimizing makespan criterion. *Expert Syst. Appl.* **2020**, *160*, 113678. [\[CrossRef\]](#)
32. Auger, A.; Hansen, N. Linear convergence of comparison-based step-size adaptive randomized search via stability of Markov chains. *SIAM J. Optim.* **2016**, *26*, 1589–1624. [\[CrossRef\]](#)
33. Hellwig, M.; Beyer, H.-G. On the steady state analysis of covariance matrix self-adaptation evolution strategies on the noisy ellipsoid model. *Theor. Comput. Sci.* **2020**, *832*, 98–122. [\[CrossRef\]](#)
34. Bolufe Rohler, A.; Tamayo Vera, D.; Chen, S. An LaF-CMAES hybrid for optimization in multi-modal search spaces. In Proceedings of the IEEE Congress on Evolutionary Computation, Donostia, Spain, 5–8 June 2017; pp. 757–764. [\[CrossRef\]](#)
35. Kumar, A.; Misra, R.K.; Singh, D. Improving the local search capability of Effective Butterfly Optimizer using Covariance Matrix Adapted Retreat Phase. In Proceedings of the IEEE Congress on Evolutionary Computation, Donostia, Spain, 5–8 June 2017; pp. 1835–1842. [\[CrossRef\]](#)
36. Zamuda, A.; Brest, J.; Mezura-Montes, E. Structured Population Size Reduction Differential Evolution with Multiple Mutation Strategies on CEC 2013 real parameter optimization. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013.
37. Liang, J.; Qu, B.; Suganthan, P.; Hernández-Díaz, A. *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization*; Technical Report; Computational Intelligence Laboratory, Zhengzhou University: Zhengzhou, China, 2013.
38. Carrasco, J.; García, S.; Rueda, M.M.; Das, S.; Herrera, F. Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm Evol. Comput.* **2020**, *54*, 100665. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.