



OPEN Surrogate-assisted global and distributed local collaborative optimization algorithm for expensive constrained optimization problems

Xiangyong Liu¹, Zan Yang^{1,2}✉, Jiansheng Liu^{1,3}, Junxing Xiong¹, Jihui Huang^{1,3}, Shuiyuan Huang⁴✉ & Xuedong Fu⁵

This paper presents a surrogate-assisted global and distributed local collaborative optimization (SGDLCO) algorithm for expensive constrained optimization problems where two surrogate optimization phases are executed collaboratively at each generation. As the complexity of optimization problems and the cost of solutions increase in practical applications, how to efficiently solve expensive constrained optimization problems with limited computational resources has become an important area of research. Traditional optimization algorithms often struggle to balance the efficiency of global and local searches, especially when dealing with high-dimensional and complex constraint conditions. For global surrogate-assisted collaborative evolution phase, the global candidate set is generated through classification collaborative mutation operations to alleviate the pre-screening pressure of the surrogate model. For local surrogate-assisted phase, a distributed central region local exploration is designed to achieve intensively search for promising distributed local areas which are located by affinity propagation clustering and mathematical modeling. More importantly, a three-layer adaptive selection strategy where the feasibility, diversity and convergence are balanced effectively is designed to identify promising solutions in global and local candidate sets. Therefore, the SGDLCO efficiently balances global and local search during the whole optimization process. Experimental studies on five classical test suites demonstrate that the SGDLCO provides excellent performance in solving expensive constrained optimization problems.

Keywords Surrogate-assisted evolutionary algorithm, Expensive constrained optimization problems, Global search, Local search

Evolutionary algorithms (EAs) such as Differential evolution (DE)¹, Teaching-Learning-Based Optimization (TLBO)², Particle swarm optimization (PSO)³, Grey wolf optimizer (GWO)⁴ have shown powerful search abilities in solving black-box problems where the gradient information of any input cannot be obtained. These algorithms not only handle single-objective optimization problems⁵ but also demonstrate significant advantages in multi-objective optimization problems (MOOs)⁶, especially when multiple conflicting objectives need to be optimized simultaneously. As the problem dimensions increase, evolutionary algorithms continue to maintain strong competitiveness in high-dimensional optimization problems⁷. Specifically, various real-world engineering problems, such as the design of hybrid renewable energy systems⁸, task scheduling for airships⁹, the open vehicle routing problem¹⁰, and microgrid design¹¹, have been successfully solved using EAs. These problems are usually unconstrained optimization problems or with bound constraints. In fact, there are many real-world engineering cases involving complex constraints, such as computational fluid dynamics¹², computational electromagnetics¹³, and antenna design¹⁴. Zhou et al.¹⁵ proposed an epsilon constraint handling method that

¹School of Advanced Manufacturing, Nanchang University, Nanchang 330031, China. ²Jiangxi Tellhow Sci-Tech Co., Ltd, Nanchang 330031, China. ³Engineering Technology Research Center of Manufacturing Informatization of Jiangxi Province, Nanchang 330031, China. ⁴School of Mathematics and Computer Sciences, Nanchang University, Nanchang 330031, China. ⁵Jiangxi Tellhow Power Technology Co., Ltd, Nanchang 330031, China. ✉email: yangzan@ncu.edu.cn; huangshuiyuan@ncu.edu.cn

simultaneously considers objective function values and constraint violations, and integrated it into a multi-objective evolutionary algorithm based on decomposition (MOEA/D)¹⁶. An adaptive fuzzy penalty method for mixed constraints optimization problems was proposed in¹⁷. Ang et al.¹⁸ employed PSO with the feasibility rule-based constraint handling method to handle constraint optimization problems. The relevant literatures about constrained optimization problems are thoroughly reviewed in¹⁹. It can be easily concluded that incorporating classical constraint handling techniques, such as penalty function method²⁰, feasibility rules²¹, stochastic ranking method²², ensemble methods²³, into EAs for biasing the search directions towards feasible region has received wide-spread attention.

However, time-consuming simulations are required to obtain the objective and constraints of many complex engineering cases, and such kind of problems where both the objective and constraints are expensive to evaluate are defined as expensive constrained optimization problems (ECOPs). As we know, EAs usually consume 5000D where *D* is the dimension of the solved problem functions evaluations to obtain satisfied or feasible solutions, which is computational unaffordable for ECOPs within an acceptable design cycle. Hence, effective surrogate models such as Kriging^{24–28}, RBF^{29–32}, SVM³³, etc., which can effectively approximate simulation problems, are used to replace unnecessary time-consuming simulation evaluations in evolutionary algorithms. This significantly accelerates the convergence speed of the algorithm. Such surrogate model-assisted evolutionary algorithms (SAEAs), widely applied in handling expensive constrained optimization problems, demonstrate excellent performance.

Generally, there are three key components in affecting the performance of SAEAs for ECOPs, such as constraint handling techniques, surrogate construction methods, and surrogate management strategies. Hence, the research progresses of SAEAs are summarized as follows from the perspectives of the three components.

Firstly, appropriate constraint-handling techniques (CHTs) can bias the search of the SAEAs towards the feasible region, thus the feasibility and convergence can be well maintained by designing effective combination between SAEAs and CHTs. Currently, several CHTs such as penalty function method²⁰, feasible rules²¹ and the stochastic ranking method²² are widely employed in SAEAs. Specifically, the penalty function method guides the algorithm into the feasible region by incorporating the degree of constraint violation into the objective function, thereby penalizing the objective value of infeasible solutions. Li et al.³⁴ explored methods for handling expensive inequality constraints within a dynamic surrogate-based optimization framework, in which the inequality constraints are handled by three different ways, such as constraining expected improvement (EI) function, penalizing surrogate prediction, and penalizing objective function. Multiple local surrogates and multiple penalty functions are designed in³⁵ where the constraints of each subproblem are penalized by adopting different penalty coefficients and search areas. Lu et al.³⁶ proposed an innovative Bayesian optimization method that effectively handles unknown equality and inequality constraints using exact penalty functions. However, penalty function method is sensitive to the penalize factor, and the value of the penalize factor is highly dependent on problems. In contrast, the advantage of stochastic ranking lies in its ability to effectively balance constraint feasibility and objective optimization through probabilistic ranking, thereby reducing the reliance on precisely designing penalty factors. Miranda-Varela et al.³⁷ studied the relationship between surrogate models and different constraint-handling techniques, and different combinations between surrogate-assisted DE and different CHTs are designed to test the performances. To address constrained combinatorial optimization problems, Gu et al.³⁸ proposed an adaptive stochastic ranking strategy for developing an efficient surrogate-assisted multi-objective particle swarm optimization algorithm. A stochastic ranking-based surrogate-assisted evolutionary algorithm was proposed in³⁹ to deal with offline data-driven optimization problems. Moreover, feasibility-based constraint handling techniques first prioritize feasible solutions by sorting them based on objective function values, and then sort infeasible solutions according to the degree of constraint violation, thereby guiding the algorithm into the feasible region. In⁴⁰, a classification collaborative mutation operation was designed to fully utilize information from both subpopulations and guide the evolutionary direction of the algorithm towards the feasible region. Wang et al.⁵ proposed a differential evolution algorithm that combines global and local surrogate models which provides an effective solution for addressing expensive constrained optimization problems, particularly those involving inequality constraints.

Secondly, the surrogate construction methods in SAEAs mainly refers to using different surrogates such as Kriging, RBF and SVM to provide guidance. Kriging is a machine learning method based on probabilistic statistics theory, which can be viewed as a stochastic process model used for interpolation. Zhang et al.⁴¹ proposed a two-stage optimization method that combines the Kriging model, in which a new Pareto dominance relationship is created by incorporating probability distribution information derived from the Kriging model. By integrating a Kriging surrogate model, a multi-stage evolutionary algorithm designed in⁴² successfully enhances the efficiency and accuracy in solving expensive multi-objective optimization problems. Song et al.⁴³ proposed a Kriging-assisted two-archive evolutionary algorithm where the information is exchanged between the two archives and the predictive capabilities of the Kriging model is used to guide the search process. By contrast, RBF is an approximation function based on the distance between input data and center points, and RBF model is widely used in SAEAs due to its excellent modeling efficiency. A distributed RBF-assisted differential evolution algorithm in⁴⁴ was designed to combine the RBF model with a distributed computing framework. Bai et al.⁴⁵ proposed a method that combines surrogate models with clustering-based sampling which effectively improves solution quality in high-dimensional expensive black-box optimization problems. Chen et al.⁴⁶ introduced an evolutionary algorithm based on the RBF surrogate model which combined the RBF model with the evolutionary algorithm. Support Vector Machine (SVM) is a supervised learning algorithm primarily used for classification, and the core idea of SVM is to find the optimal hyperplane that maximizes the margin between classes, thereby improving classification accuracy and robustness. Horaguchi et al.⁴⁷ explored a classification-based multi-objective evolutionary algorithm (MOEA) for high-dimensional expensive optimization problems, incorporating dimensionality reduction techniques to improve optimization efficiency. A design optimization

method that combines sequential radial basis functions and support vector machines was proposed in⁴⁸ which provides an effective solution for handling expensive objective functions. Liao et al.⁴⁹ proposed a decomposition-based enhanced multi-objective evolutionary algorithm that utilizes Support Vector Machine to predict and optimize the search process.

Thirdly, surrogate management strategies mainly refer to the combination ways between surrogates and evolutionary algorithms, and they can be generally categorized into three types: individual-based strategy, generation-based strategy, and population-based strategy. For the first one, the surrogate model is used to provide predictions for many candidate solutions, and the identified several promising solutions are evaluated by the truly expensive evaluation. Lin et al.⁵⁰ proposed a classification model-based assisted preselection and environment selection approach which provides an effective framework for solving expensive optimization problems. Regis et al.⁵¹ used the RBF model to find the most promising individual and improve the current overall best individual by finding the global minimum of the surrogate model. Wang et al.⁵ proposed a differential evolution algorithm where both global and local surrogates were collaboratively employed. The generation-based strategy refers to using surrogate models for fitness evaluation in some generations of the evolutionary process, while using the truly expensive function or other surrogate models for fitness evaluation in other generations. The purpose of this approach is to reduce computational costs by decreasing the number of evaluations with the surrogate model, while retaining the ability to use the truly expensive function for precise evaluation at critical moments, thus balancing computational efficiency and optimization accuracy. In⁵², the global RBF model is restarted every few generations and the population is re-initialized at each restart using the best sample points. Zheng et al.⁵³ adopted a noise-based model switching strategy to select a surrogate model that meets the requirements from GP and RBFN for each generation. In different stages of the algorithm, Yu et al.⁵⁴ used coarse Gaussian process model and the fine radial basis function model respectively to assist the differential evolution algorithm. For population-based strategies, the selection and management of surrogate models are dynamically adjusted based on the current state of the population. This involves dynamically choosing suitable surrogate models to support global exploration or local optimization, and updating the models in real-time. Li et al.⁵⁵ generated and updated surrogate by different techniques for different sub-populations and selected the suitable surrogates for each sub-population. In⁵⁶, a variable search region-based adaptive dynamic surrogate-assisted evolutionary computation method was introduced. As the current local optimal solution moves, the local search space is continuously transformed and the surrogate model is reconstructed. Couckuyt et al.⁵⁷ created a subpopulation for each surrogate model and dynamically selected the best model type based on the data available to date.

Also, there are roughly three kinds of SAEAs in solving ECOPs, i.e., global surrogate-assisted^{40,52}, local surrogate-assisted³⁵, and the collaborative between global and local surrogates⁵. For the first category, the surrogate is built by using all the evaluated solutions, then the search of EAs can be guided globally. This kind of algorithm provide powerful abilities in locating unexplored promising regions, but they are unable to achieve intensively local exploit on specific local regions. To alleviate this limitation, some algorithms introduce perturbation strategies that apply small random perturbations to the current solution, thereby exploring a larger solution space. Perturbation helps maintain diversity in local search and prevents premature convergence^{58,59}. However, it should be noted that these local surrogate-assisted optimization algorithms (especially those incorporating perturbations) may still not fully address complex ECOPs where many local optima exist in their landscapes. Therefore, for balancing the global exploration and local exploitation, the collaborative algorithms between global and local surrogates are widely devised where the global surrogate helps locate new promising regions and local surrogate accelerates the convergence in these located regions.

Based on the discussions, the existing SAEAs have indeed enhanced the optimization performance on solving ECOPs, but they cannot explicitly consider the differences of coverage regions for different solutions when executing local searches, or they don't consider the similarity of surrogates constructed for different solutions in local searches. Thus, the corresponding local regions or surrogates of many local searches may be extremely similar, since their corresponding central solutions are closer to each other as the iterations progresses. Therefore, how to efficiently allocate well-distributed local searches during the optimization process is one of the research focuses of this paper. Moreover, these existing SAEAs mainly select the truly offspring through designing single-layer screening strategy constructed based on predicted objective or constraint values, in which the diversity among different offspring individuals may be ignored to a certain extent, thus the possibility of falling into local optima may not be effectively reduced. Therefore, two important search directions emerged in our mind, i.e., how to effectively design suitable fitness function on evaluating the quality of candidate solutions where the diversity and objective or constraint values can be comprehensively considered, and how to design effective multi-layer screen strategy to select promising candidate solutions layer by layer.

Therefore, this paper designs a surrogate-assisted global and distributed local collaborative optimization (SGDLCO) algorithm for expensive constrained problems. Specifically, for the aforementioned first search focuses, a distributed central region local exploration (DCRL) is designed to diversify local searches where the affinity propagation clustering⁶⁰ and mathematical modeling are utilized to identify well-distributed central solutions. By minimizing redundant local searches for similar solutions or surrogates, the efficiency of exploitation is substantially improved. Subsequently, for the second point, the fitness functions such as diversity-based objective and constraints are designed to predict the overall quality of each candidate offspring individuals. Moreover, a three-layer adaptive screening strategy (TLAS) where the diversity, feasibility and overall quality are sequentially considered is proposed to progressively identify the truly promising offspring individual, thus the possibility of premature or trapping into local optima may be decreased since three key indicators are well balanced for each generation. Furthermore, for further improving the quality of candidate offspring individuals, a classification-cooperation mutation operation is employed in this paper to achieve the collaborative mutation between good and bad solutions.

Based on the considerations mentioned above, the main contributions of this paper are summarized as follows:

- 1) In the global surrogate-assisted collaborative evolution phase, the global candidate set is generated through classification collaborative mutation operations (CCMO) to alleviate the pre-screening pressure of the surrogate model. In CCMO, the positive guidance information of better solutions and the negative guidance information of worse solutions are fully utilized to improve the quality of candidate solutions.
- 2) In the local surrogate-assisted phase, a distributed central region local exploration (DCRLE) is designed to generate local candidate set for achieving intensively search in promising distributed local areas which are located by affinity propagation clustering and mathematical modeling. Thus, the redundant local searches for similar central solutions or surrogates can be largely alleviated, and the efficiency can be maintained by allocating specific local searches on well-distributed central solutions.
- 3) A three-layer adaptive selection strategy, which incorporates the diversity among different offspring individuals into the screening process, is adopted to progressively select promising candidate offspring solutions. Concretely, two diversity-based fitness functions where the feasibility, diversity and convergence are effectively balanced are formulated to screen candidate solutions from the global or local candidate sets.
- 4) Overall and systematic experiments on benchmark problems such as IEEE CEC2006⁶¹, and IEEE CEC2010⁶² and IEEE CEC 2017⁶³ have shown that the proposed SGDLCO provides highly competitive performance.

The structure of this paper is as follows: In Sect. 2, the problem statement and relative background techniques are presented. Section 3 describes the proposed SGDLCO. The experimental results are analyzed and summarized in Sect. 4. Conclusions are drawn in Sect. 6.

Problem statement and relative background techniques

Problem statement

The expensive constrained optimization problem with inequality constraints studied in this paper can be represented as:

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{subject to } g_j(x) \leq 0 \quad j = 1, 2, \dots, q \end{aligned} \quad (1)$$

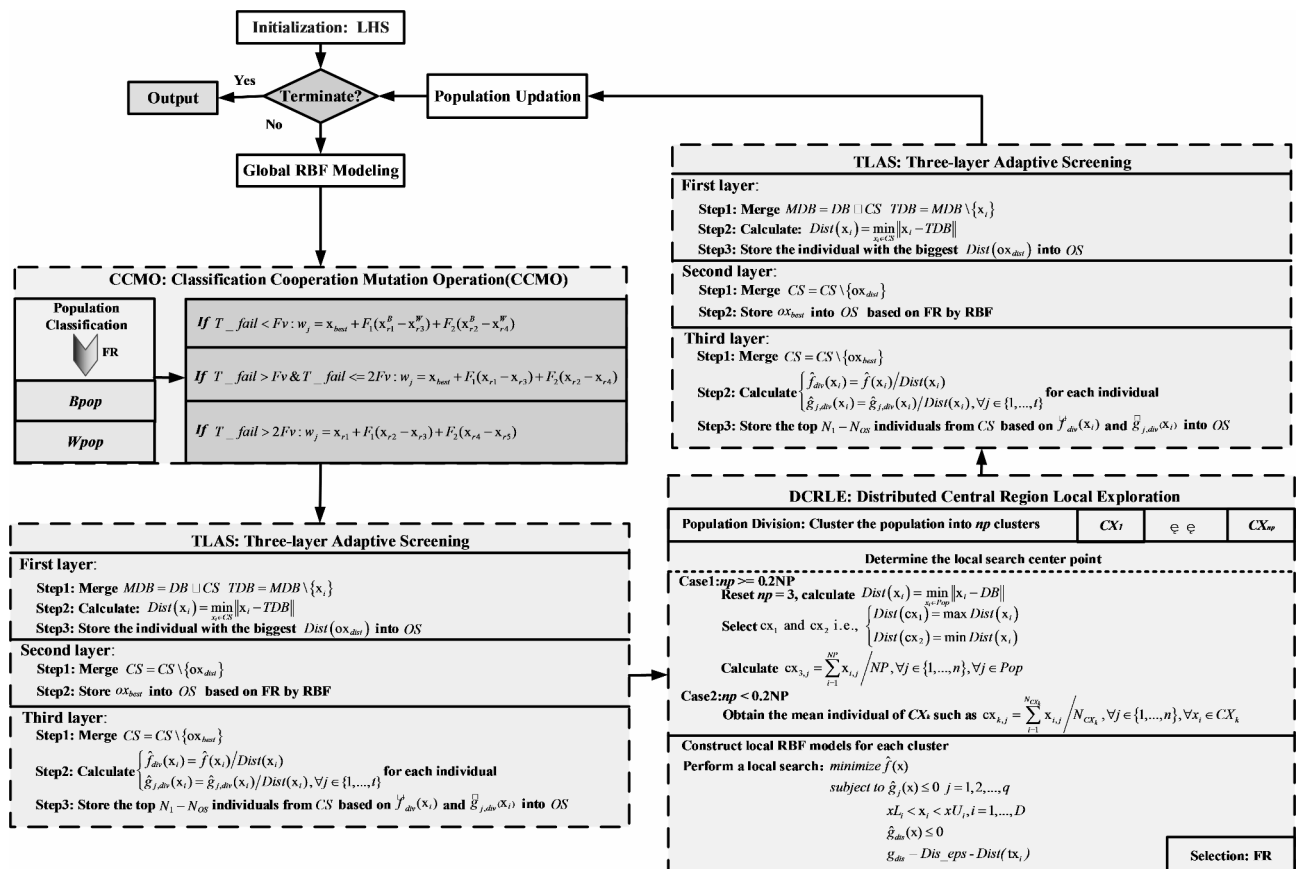


Fig. 1. Flowchart of SGDLCO.

where $f(x)$ and $g_j(x)$ denotes the objective and j -th inequality constraint functions where their responses can only be obtained by calling computationally expensive simulations, and q is the number of inequality constraints. $x = (x_1, x_2, \dots, x_D)$ is the decision vector, D is the number of decision variables, $L = (l_1, l_2, \dots, l_D)$ and $U = (u_1, u_2, \dots, u_D)$ are the lower and upper bounds of design variables. The degree of constraint violation of x can be computed as follows:

$$G_j(x) = \max(0, g_j(x)), j = 1, \dots, p \quad (2)$$

Afterward, the overall constraint violation function can be formulated as:

$$G(x) = \sum_{j=1}^p G_j(x) \quad (3)$$

Therefore, the objective of the optimization problem is to find the optimal solution that minimizes the objective function while satisfying the constraint violation function $G(x)$.

Relative background techniques

Differential evolution (DE)

DE is a population-based optimization algorithm that relies on swarm intelligence theory⁶⁴. It maintains the population by generating new individuals through mutation, selection, and crossover operations. By performing operations on the population of candidate solutions, DE explores the solution space through information exchange among different individuals in the population.

DE utilizes NP parameter vectors of dimension D , treating them as the population for each generation. Each individual is represented as:

$$x_i = (x_{i,1}, \dots, x_{i,D}) \quad (4)$$

For each individual in the population $x_i = (x_{i,1}, \dots, x_{i,D})$, the mutant vector is generated as follows:

(1) DE/rand/1

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

(2) DE/best/1

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

(3) DE/best/2

$$v_i = x_{best} + F \cdot (x_{r1} - x_{r2}) + F \cdot (x_{r3} - x_{r4}) \quad (7)$$

(4) DE/current-to-best/1

$$v_i = x_i + F \cdot (x_{best} - x_i) + F \cdot (x_{r1} - x_{r2}) \quad (8)$$

In the above formula, $r1, r2, r3$, and $r4$ are four completely different random numbers which is selected from $[1, i] \cup (i, NP]$ and NP must be greater than 4. x_{best} presents the optimal solution in the population. F is a scaling factor that amplifies the deviation variable. By performing crossover operation to increase the diversity of perturbed parameter vectors, the trial vector becomes:

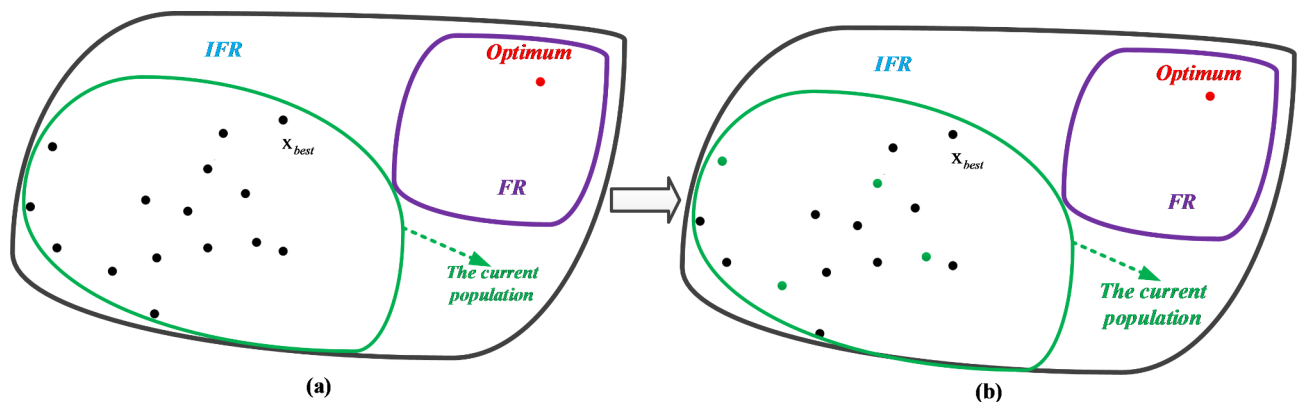


Fig. 2. Randomly select four individuals from the initial population.

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand_j \leq CR \text{ or } j = j_{rand} \\ x_{i,j} & \text{otherwise} \end{cases}, \quad j = 1, \dots, n \quad (9)$$

where $rand_j$ presents the j -th estimate of a random number generator producing values between $[0, 1]$, j_{rand} is a randomly chosen integer from $\{1, \dots, n\}$, CR represents the crossover operator, which ranges from $[0, 1]$.

Radial basis function (RBF)

RBF is a function whose output value depends solely on the distance between the input and a certain center point. RBF approximates complex nonlinear relations by introducing a linear combination of

local basis functions in the input space. RBF demonstrates excellent universal approximation capabilities and performs exceptionally well in handling nonlinear problems. Given the existing data $\{x_i, y_i | x_i \in R, i = 1, \dots, N\}$, the RBF approximation can be expressed as follows:

$$\hat{f}(x) = w^T \varphi = \sum_{i=1}^N \omega_i \phi(\|x - x_i\|) \quad (10)$$

where x_i represents the computational center point of the i th basis function among all basis functions, $\|\bullet\|$ represents the distance between the input x and x_i , N is the number of center points. ω_i is the weight coefficient, Φ is the Gram matrix which can be defined in Eq. (11).

$$\Phi = \begin{bmatrix} \varphi(\|x_1 - x_1\|) & \cdots & \varphi(\|x_1 - x_N\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|x_N - x_1\|) & \cdots & \varphi(\|x_N - x_N\|) \end{bmatrix} \quad (11)$$

Affinity propagation clustering (APC)

Affinity Propagation Clustering (APC) is an advanced unsupervised clustering algorithm developed by Frey and Dueck in 2007⁶⁰. This technique automatically determines the number of clusters by identifying cluster centers through iterative message passing. The algorithm operates on a similarity matrix, where each element represents the negative squared distance between data points. There are two kinds of messages involved in APC, they are responsibility $r(i, k)$ and availability $s(i, k)$. $r(i, k)$ can be computed as follows:

$$r(i, k) \leftarrow s(i, k) - \max_{k' \neq k} \{a(i, k') + s(i, k')\} \quad (12)$$

where $s(i, k)$ is the similarity between i and k , and $a(i, k')$ is the availability of k' .

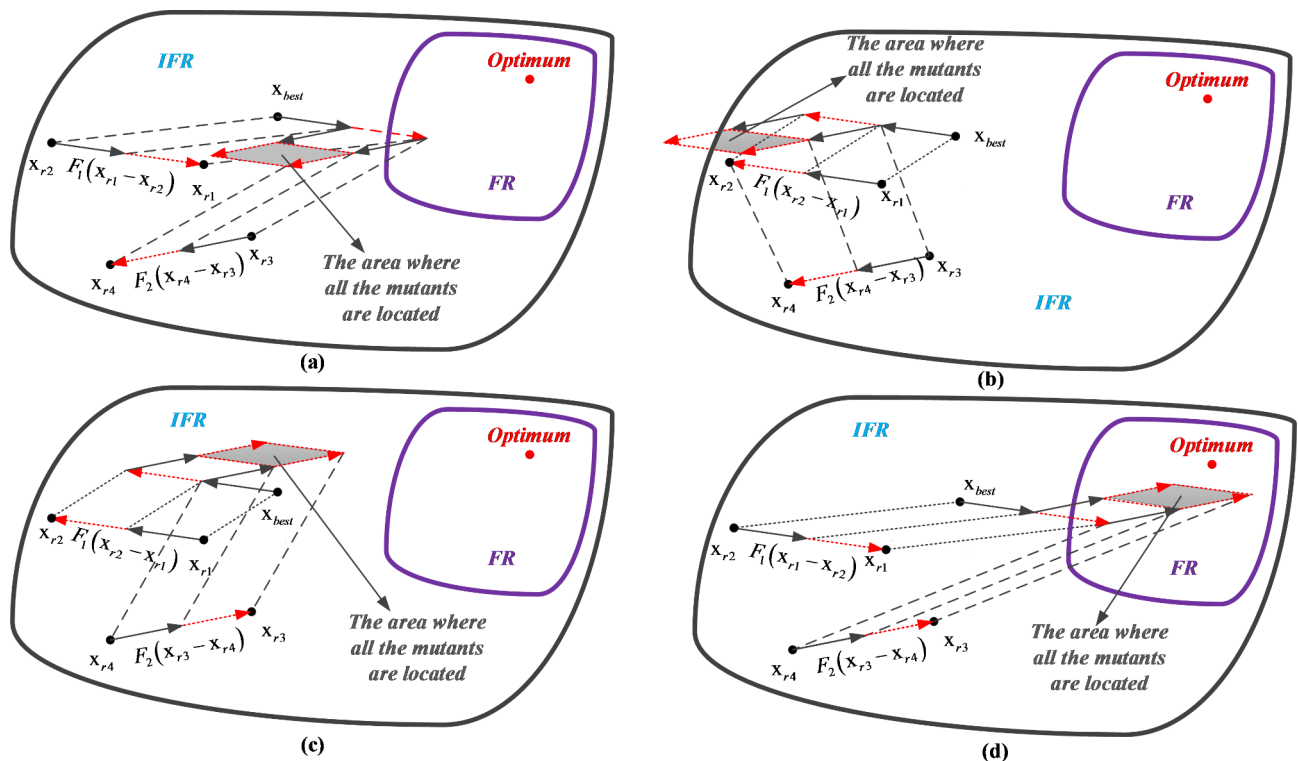


Fig. 3. Four cases of DE/best/2.

$a(i, k)$ can be computed as follows:

$$a(i, k) \leftarrow \min\{0, r(k, k) + \sum_{i' \notin \{i, k\}} \max\{0, r(i', k)\}\} \quad (13)$$

These two messages are iteratively updated till convergence, then the clustering result can be obtained by:

$$\hat{c}_i = \arg \max_k \{a(i, k) + r(i, k)\} \quad (14)$$

Proposed algorithm

Most existing self-adaptive evolutionary algorithms (SAEAs) use traditional differential evolution mutation operations to generate offspring individuals when dealing with expensive optimization problems with inequality constraints. This method has a significant degree of blindness in producing offspring individuals. Therefore, this paper employs the classification cooperation mutation operation (CCMO) during the global search phase to generate offspring individuals, thereby effectively utilizing limited computational resources. As the algorithm progresses, the search space gradually narrows. The range of local search often becomes excessively repetitive, which can significantly waste computational resources. Therefore, this paper designs a distributed central region local exploration (DCRLE) to avoid the redundancy in local search. The existing SAEAs primarily select truly promising offspring through a single-layer screening strategy based on designed predictive objectives or constraint values. However, the diversity among different offspring individuals may be somewhat overlooked, which could limit the effectiveness of reducing the likelihood of falling into local optima. While the current SAEAs have improved optimization performance for solving ECOPs by incorporating specific local search strategies, the center solutions corresponding to local searches for each population individual become increasingly similar as iterations progress, and thus many redundant local searches for similar solutions or surrogates are designed in these SAEAs. Therefore, to filter genuinely potential offspring solutions, a fitness function based on diversity, feasibility, and overall quality is designed to assess the overall quality of each candidate offspring individual. Additionally, a three-layer adaptive screening strategy (TLAS) is proposed, which sequentially considers diversity, feasibility, and overall quality. To diversify local searches, this paper introduces a distributed center region local exploration, and the affinity propagation clustering⁶⁰ and mathematical modeling are designed to identify well-distributed center solutions. This significantly reduces redundant local search iterations for similar solutions or alternatives.

SGDLCO Method

The flow chart of SGDLCO is shown in Fig. 1. In SGDLCO, the global and local surrogate-assisted phases are collaboratively executed where the global surrogate is employed to provide predictions for the global candidate solution sets generated by the classification collaborative mutation operation, and the local surrogate is used to build the distributed local search problems constructed for well-distributed central solutions selected based on affinity propagation clustering and mathematical modeling. More importantly, for effectively balancing feasibility, diversity and convergence, the three-layer adaptive screening strategy is designed to screen high-promising solutions from global and local candidate sets which are respectively generated in global and local

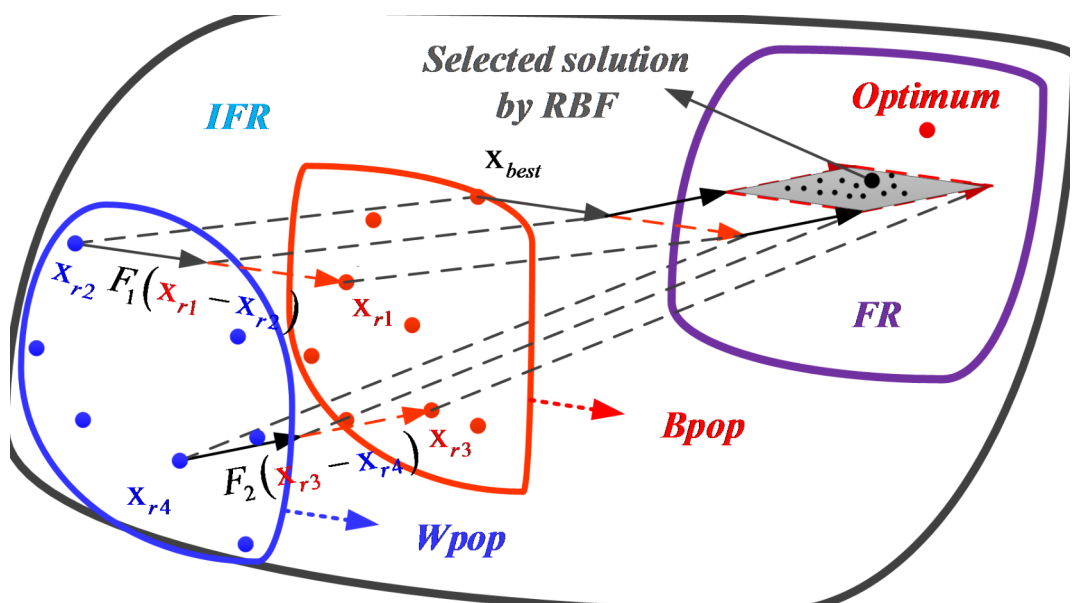


Fig. 4. The tendency of movement of mutant vectors when all solutions are infeasible.

surrogate-assisted phases. Thus, the proposed algorithm maintains efficient collaborative cooperation between global and local searches at each generation.

The main framework of SGDLCO is shown in Algorithm 1. The initial population Pop is generated by Latin hypercube sampling (LHS)⁶⁵. Then, the objective function and constraint functions of all individuals in the initial population are evaluated, and the obtained results are stored in the database DB . For global surrogate-assisted collaborative evolution phase, the global RBF model is constructed by using all the individuals in DB . For each parent solution, the offspring candidate V_k is produced through the classification cooperation mutation operation (CCMO), as detailed in Sect. 3.2. Gx is selected by applying the three-layer adaptive screening (TLAS) as described in Sect. 3.3 for candidate set V_k . Subsequently, Gx is evaluated by the real objective and constraint functions, and the DB is updated. Pop is updated by selecting the top NP individuals from the merged population of Pop and Gx . During local surrogate-assisted phase, the candidate set LV is generated by the distributed central region local exploration (DCRLE) as explained in Sect. 3.4. Similarly, the TLAS is used to select Lx , and then update DB and Pop . The above procedure is repeated until the maximum number of fitness evaluations is reached.

Algorithm 1 SGDLCO method	
Input:	Population size: NP . The maximum number of function evaluations: $MaxFes$. The database: DB .
Output:	The optimized solution: x_{best} .
1.	Generate the initial population, $Pop = \{x_1, x_2, \dots, x_{NP}\}$, by Latin hypercube sampling.
2.	Evaluate the initial population and archive them into DB , and set $Fes = NP$.
3.	While $Fes \leq MaxFes$
4.	// The global surrogate-assisted collaborative evolution phase
5.	Construct global RBF models, i.e., $\hat{f}^G, \hat{g}_1^G, \dots, \hat{g}_t^G$, for objective and constraints by using all the individuals in DB .
6.	Generate candidate set, i.e., V_k by the classification cooperation mutation operation (CCMO) shown in Algorithm 2.
7.	Conduct Algorithm 3 (called Three-Level Adaptive Screening, TLAS) to obtain the offspring set, i.e., Gx from $GU = \{u_1, \dots, u_{NP}\}$.
8.	Evaluate Gx , and store them into DB , and $Fes = Fes + N_{Gx}$. // N_{Gx} donates the number of individuals in Gx .
9.	Select the best NP individuals such as Pop from the merged population based on FR, i.e., $MPop = Pop \cup Gx$.
10.	// The local surrogate-assisted phase
11.	Generate candidate set, i.e., LV by the distributed central region local exploration (DCRLE) as explained in Algorithm 4.
12.	Conduct Algorithm 3 (called Three-Layer Adaptive Screening, TLAS) to obtain the offspring set, i.e., Lx from LV .
13.	Evaluate Lx , and store them into DB , and $Fes = Fes + N_{Lx}$. // N_{Lx} donates the number of individuals in Lx .
14.	Select the best NP individuals such as Pop from the merged population based on FR, i.e., $MPop = Pop \cup Lx$.
15.	End While
16.	Select the best solution x_{best} from Pop based on FR.

Classification Cooperation Mutation Operation (CCMO)

Most of the existing SAEAs for handling expensive optimization problems with inequality constraints adopt traditional mutation operations of evolutionary algorithms to generate offspring individuals, which introduces a significant degree of randomness. Therefore, the classification-collaboration mutation operation designed in⁴⁰ is adopted in this section to design the mutation operation. For visualizing the search behavior of the designed CCMO, the classic DE/best/2 in DE is used as an example to illustrate the randomness of classical mutation. The DE/best/2 primarily generates mutated individuals by adding four random differential vectors to the current best individual, thereby creating a mutation direction. In this context, x_{best} represents the current optimal individual in the optimization process, while x_{r1}, x_{r2}, x_{r3} , and x_{r4} are four individuals randomly selected from the current population, F_1 and F_2 are the scaling factors.

As shown in Fig. 2(a), the entire area inside the black line represents the overall design domain, FR and IFR represent feasible and infeasible regions respectively. Within this design domain, the area inside the purple line is the feasible region, while the remaining areas are infeasible region. The current population consists of 15 individuals, all represented as black dots. Thus, we first carry out the initial step of the DE/best/2 mutation by randomly selecting four different differential vectors. As shown in Fig. 2(b), the four green dots correspond to these four differential vectors. After determining the positions of the four differential vectors, it is necessary to further establish the selection order of these vectors and the values of the scaling parameters to generate mutated individuals. Due to the randomness of the mutation strategy, there are a total of 24 possible selection orders based on permutations. For simplicity, we will only show the visualization of the four cases corresponding to the limited rules, such as the first differential vector term consisting of x_{r1} and x_{r2} , and the second differential vector term is composed of x_{r3} and x_{r4} . Then the four cases are shown as below:

- (1) in Fig. 3(a), the mutation operation can be rewritten as: $v_i = x_{best} + F_1 \cdot (x_{r1} - x_{r2}) + F_2 \cdot (x_{r4} - x_{r3})$
- (2) in Fig. 3(b), the mutation operation can be rewritten as: $v_i = x_{best} + F_1 \cdot (x_{r2} - x_{r1}) + F_2 \cdot (x_{r4} - x_{r3})$
- (3) in Fig. 3(c), the mutation operation can be rewritten as: $v_i = x_{best} + F_1 \cdot (x_{r2} - x_{r1}) + F_2 \cdot (x_{r3} - x_{r4})$
- (4) in Fig. 3(d), the mutation operation can be rewritten as: $v_i = x_{best} + F_1 \cdot (x_{r1} - x_{r2}) + F_2 \cdot (x_{r3} - x_{r4})$

In Fig. 3, the red dashed arrows indicate the diversity of mutation directions resulting from different values of the scaling factors. The gray shaded areas correspond to the regions occupied by all mutated individuals. From

these four subfigures, it can be observed that only the case shown in Fig. 3(d) leads all mutated individuals into the feasible region, while the mutated individuals in the other three cases are far from the feasible region. Therefore, in traditional mutation operations, the randomness in the selection of differential vectors leads to significant blindness in the mutation directions. When the surrogate model has some degree of prediction error, it becomes inefficient in filtering out a few high-potential individuals from a large pool of candidate offspring. Therefore, ensuring that each mutation is based on the situation in Fig. 3(d) to produce these high-potential individuals is the key focus of this section. In summary, this can be achieved by adjusting the selection rules for the four differential vectors in DE/best/2 to consistently obtain high-potential mutated individuals from the initial population shown in Fig. 2(b). Based on the visualization and the mutation formula in Fig. 3(d), we can derive two conclusions:

- (1) x_{r1} and x_{r3} are relatively close to the feasible region, and they produce a positive guidance effect.
- (2) x_{r2} and x_{r4} are relatively far from the feasible region, and they produce a negative guidance effect

Therefore, for fully utilizing the positive guidance information of better solutions and the negative guidance information of worse solutions, the population is classified into two subpopulations based on feasibility rules in²¹, in which the first subpopulation consists of better solutions and the other is composed of the rest solutions. After this, the candidate set is generated for each parent solution by collaborative cooperation between the two subpopulations, and a high-promising candidate individual is selected from this candidate set based on RBF predictions. The pseudo-code of classification cooperation mutation operation (CCMO) is shown in Algorithm 2.

In Algorithm 2, the advantage of selecting DE/best/2 as the mutation operator is that it combines information from both the better and worse solutions in the current population. This enables the mutation process to move towards promising regions indicated by the best solutions while avoiding unfavorable areas. Therefore, DE/best/2 is well-suited to construct the CCMO shown in lines 5–6 of Algorithm 2. Specifically, two different solutions such as x_{r1}^B and x_{r3}^B , which are randomly selected from $BPop$, are utilized to determine the positive guidance locations of CCMO; while x_{r2}^W and x_{r4}^W are selected from $WPop$ to locate the negative guidance. However, incorporating massive current greedy information to mutation may lead the algorithm into local optima more easily. Hence, different cooperative ways are employed to adjust the evolution of the population based on its current state, which is shown in lines 4–11. Concretely, the current greedy information brought by the better solutions or the best solutions is gradually diluted when the algorithm presents instability or stagnation, i.e., the number of failures reaches the predetermined value.

Algorithm 2 Classification Cooperation Mutation Operation (CCMO)	
Input:	The population: Pop . Threshold for determining which mutation strategy to use: Fv . The count of failed attempts to update the population's best solution: T_{fail} .
Output:	V_k : Candidate set.
1.	Divide the Pop into two subpopulations based on FR where the $BPop$ contains the top half individuals, and the $WPop$ contains the rest.

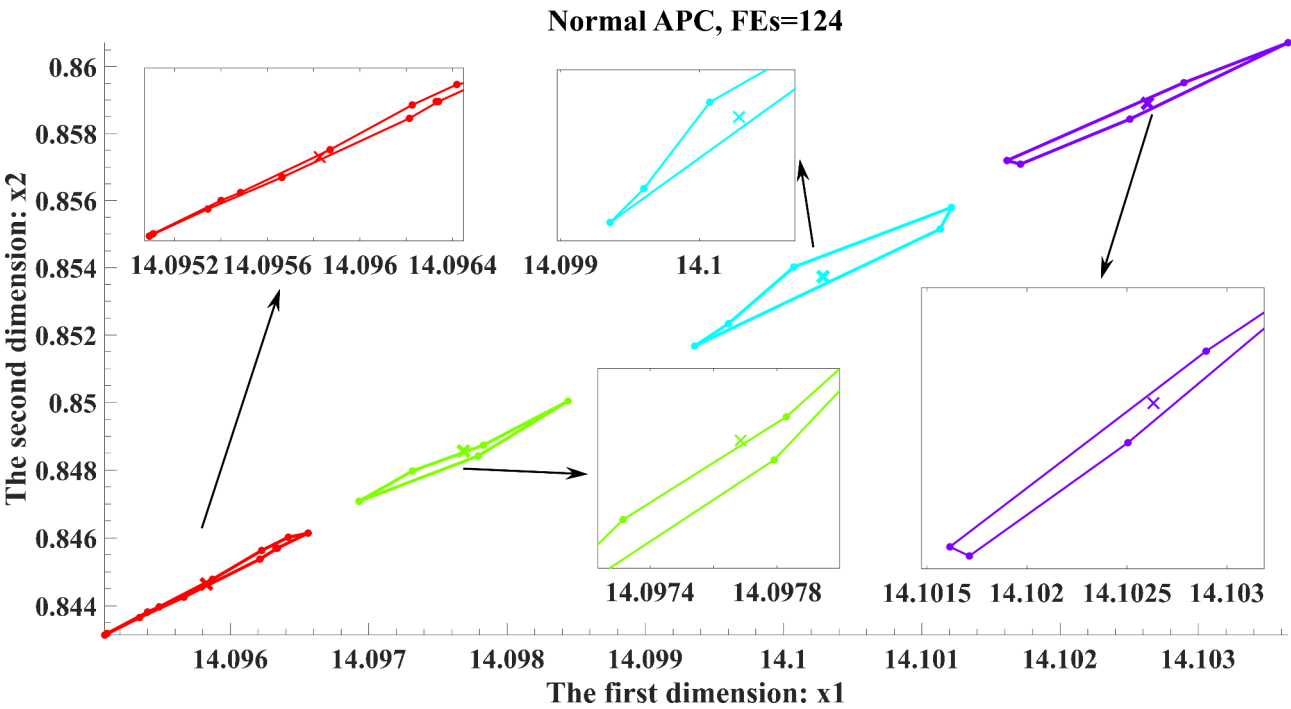


Fig. 5. The situation of normal APC.

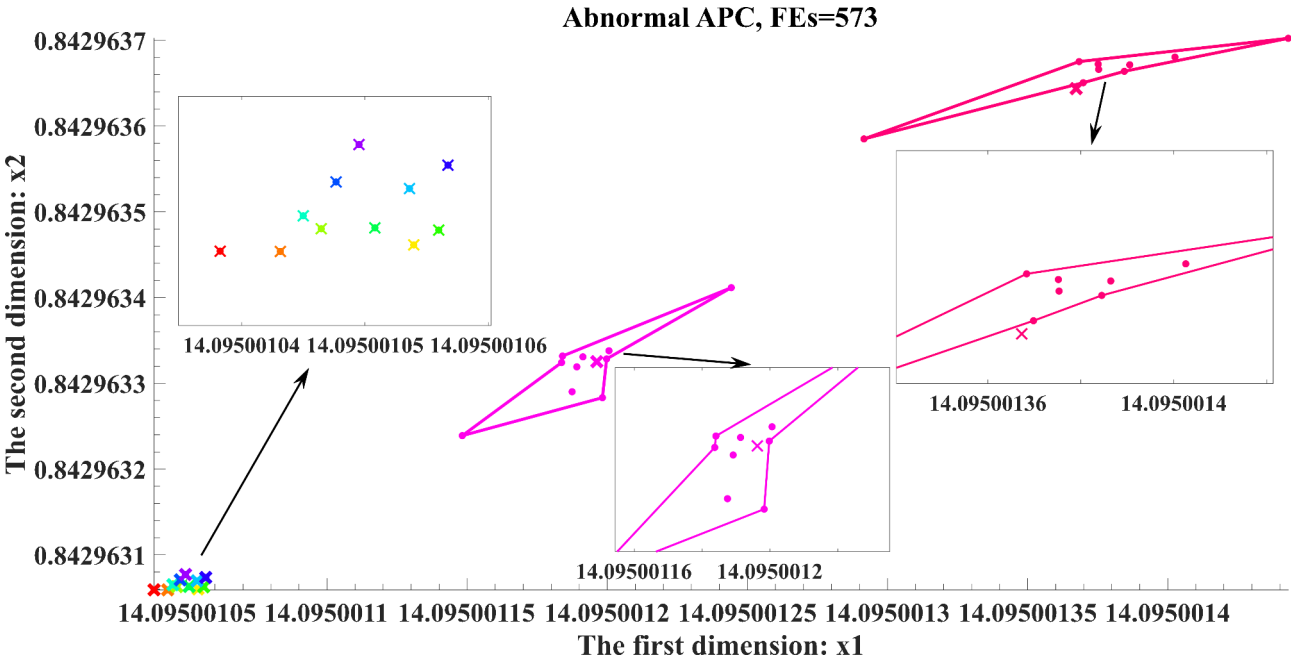


Fig. 6. The situation of excessive APC.

Algorithm 2 Classification Cooperation Mutation Operation (CCMO)	
2.	For $i = 1 : NP$
3.	For $j = 1 : NC$
4.	If $T_fail < Fv$
5.	Randomly select two distinct individuals x_{r1}^B and x_{r2}^B from $BPop$, x_{r3}^W and x_{r4}^W from $WPop$.
6.	$w_j = x_{best} + F_1(x_{r1}^B - x_{r3}^W) + F_2(x_{r2}^B - x_{r4}^W)$. // $F_1 = (1 + rand(0, 1))/2$, $F_2 = (1 + rand(0, 1))/2$
7.	Else If $T_fail > Fv$ & $T_fail \leq 2Fv$
8.	$w_j = x_{best} + F_1(x_{r1} - x_{r2}) + F_2(x_{r3} - x_{r4})$, x_{r1} , x_{r2} , x_{r3} , and x_{r4} are four distinct individuals selected from Pop .
9.	Else If $T_fail > 2Fv$
10.	$w_j = x_{r1} + F_1(x_{r2} - x_{r3}) + F_2(x_{r4} - x_{r5})$, x_{r1} , x_{r2} , x_{r3} , x_{r4} and x_{r5} are five distinct individuals selected from Pop .
11.	End If
12.	Obtain u_j by conducting binomial crossover on w_j and x_i .
13.	End For
14.	Select the best one such as v_k from U_k based on FR by using the global RBF predictions on U_k , and $V_k = V_k \cup \{v_k\}$. // $U_k = \{u_1, \dots, u_{NC}\}$.
15.	End For

As the algorithm progresses, the proportion of feasible solutions in the population gradually increases, while the proportion of infeasible solutions gradually decreases. The situation in which all the solutions are infeasible is selected as the example to clearly present the trajectory of mutation vectors, the tendency of movement of mutant vectors is shown in Fig. 4. In Fig. 4, the population Pop is divided into $BPop$ and $WPop$. $BPop$ includes the top half solutions of Pop , and $WPop$ is composed of the remaining part. It can be easily found that the $BPop$ is closer to the feasible region compared to $WPop$. This indicates that the generated candidate solutions should be moved towards the positive directions located by $BPop$, and the negative directions of $WPop$ can be also used as guidance to further move the candidate solutions towards feasible region. Specifically, $x_{r1,g}$ and $x_{r2,g}$ are two randomly selected solutions from $BPop$, $x_{r3,g}$ and $x_{r4,g}$ are two randomly selected solutions from $WPop$. The line connecting $x_{r1,g}$ to $x_{r2,g}$, and the line connecting $x_{r3,g}$ to $x_{r4,g}$ represent the direction and length of $F_1 \cdot (x_{r1,g} - x_{r2,g})$ and $F_2 \cdot (x_{r3,g} - x_{r4,g})$ respectively. Hence, All the possible cases that $F_1 \cdot (x_{r1,g} - x_{r2,g})$ can be extended are on the red dotted line between $x_{r1,g}$ and $x_{r2,g}$. The red dotted line between $x_{r3,g}$ and $x_{r4,g}$ contains all the possible cases that $F_2 \cdot (x_{r3,g} - x_{r4,g})$ can be extended. Therefore, the parallelogram area formed by vertices $w_{i,g}^1$, $w_{i,g}^2$, $w_{i,g}^3$, and $w_{i,g}^4$ represents the region where the candidate offspring solutions are located. Hence, although all the solutions in current population are infeasible, all of the generated candidate solutions are moved into feasible region. This directly demonstrate that CCMO is capable of generating high-promising candidate solutions, then the screen pressure of RBF can be greatly decreased.

Three-layer adaptive screening (TLAS)

During the algorithm’s iterations, the selection strategy often becomes too stringent and only allows the optimal solutions to survive, the diversity among different offspring individuals may be ignored to a certain extent. This excessive selection pressure reduces the population’s diversity, increasing the risk of falling into local optima. When the population tends to consist of highly similar individuals, the algorithm is prone to getting stuck in local optima and fails to effectively find global optima. This situation can lead to a homogenized search process, resulting in premature convergence. Therefore, maintaining a certain level of diversity is crucial to promote a more comprehensive search and innovation, avoiding premature convergence to local optima. By introducing diversity maintenance mechanisms, the adaptability of the algorithm can be enhanced, increasing the chances of finding better solutions. To alleviate the selection pressure on the population, as shown in Algorithm 3, a three-layer adaptive screening strategy (TLAS), which incorporates the diversity among different offspring individuals into the screening process, is adopted to progressively select promising candidate offspring solutions.

Specifically, TLAS sequentially considers the diversity, feasibility, and overall quality of each individual, and selects offspring based on these three indicators.

The first layer of the screening function is shown in lines 2–3 of Algorithm 3. The minimum distance calculated by Eq. (15) represents the shortest distance between an individual and all other individuals in *TDB*. For each individual in the candidate *CS*, the minimum distance $Dist(x_i)$ from the individual to all individuals in the total database *TDB* except itself is calculated. As the iterations progress, the individuals in the population gradually become homogeneous. In fact, the individual with the biggest minimum distance to all other individuals indicates that it is located in a relatively isolated region in the design space. This means that the individual exhibits greater diversity from other individuals. Therefore, the individual with the biggest $Dist(x_i)$ is selected. It represents relatively unique positions in the design space, which helps prevent the algorithm from falling into local optima.

$$Dist(x_i) = \min_{x_i \in CS} \|x_i - TDB\|$$
 (15)

The second layer of the screening function is shown in line 4 of Algorithm 3. After selecting the individual with diversity as described above, the individual with the best predicted value in the current population is selected based on FR. Such individual plays a role in guiding the evolution direction of the population, and the region where the individual resides represents a more promising area in the sample space.

Given the condition that the individual selected based on diversity may not has better quality measured by the predicted objective and constraints, and the individual selected based on the combined quality of the predicted objective and constraints cannot reflect the differences between individuals in the population. Therefore, the third layer screening strategy is designed for selecting the individual with overall quality as shown in lines 7–8 of Algorithm 3.

A mathematical model for screening individuals based on their overall quality is designed, as shown in Eq. (16):

$$\begin{aligned} \hat{f}_{div}(x_i) &= \hat{f}(x_i) / Dist(x_i) \\ \hat{g}_{j,div}(x_i) &= \hat{g}_{j,div}(x_i) / Dist(x_i) \end{aligned}$$
 (16)

where $\hat{f}(x_i)$ and $\hat{g}_{j,div}(x_i)$ respectively represents the predicted values of objective and *j*-th constraint functions for x_i . Then $\hat{f}_{div}(x_i)$ donates diversity-based objective, which represents the combined quality between objective and diversity; $\hat{g}_{j,div}(x_i)$ donates diversity-based constraint, which calculates the combined quality between constraints and diversity. Then the FR is employed to select a high-promising solution in terms of the overall quality of diversity, feasibility and diversity.

In summary, TLAS select individuals from the current population based on their diversity, feasibility, and comprehensive quality, aiming to maintain the quality of population individuals while generating a more exploratory population. Consequently, TLAS significantly increases the chances of finding global optimal solutions in the solution space rather than getting trapped in local optima.

Algorithm 3 Three-Layer Adaptive Screening (TLAS)	
Input:	Candidate set: <i>CS</i> (<i>GU</i> or <i>LV</i>). The Database: <i>DB</i> . The lower bound: <i>xl</i> . The upper bound: <i>xu</i> . The dimensions of an individual: <i>n</i> . The number of individuals which is selected: <i>N</i> ₁ .
Output:	Offspring set: <i>OS</i> (<i>Gx</i> or <i>Lx</i>).
1.	Construct the merged database, i.e., <i>MDB</i> = <i>DB</i> ∪ <i>CS</i> , and initialize <i>OS</i> = ∅.
2.	For each <i>x_i</i> in <i>CS</i> , <i>TDB</i> = <i>MDB</i> \ { <i>x_i</i> } and calculate the minimum distance $Dist(x_i) = \min_{x_i \in CS} \ x_i - TDB\ $.
3.	Store an individual such as <i>ox_{dist}</i> from <i>CS</i> with the biggest <i>Dist</i> (<i>ox_{dist}</i>) into <i>OS</i> .
4.	Archive the best one such as <i>ox_{best}</i> into <i>OS</i> from <i>CS</i> based on FR by using the RBF predictions on <i>CS</i> .
5.	<i>CS</i> = <i>CS</i> \ { <i>ox_{best}</i> }, and randomly select an individual such as <i>ox_{rand}</i> from <i>CS</i> , and <i>OS</i> = <i>OS</i> ∪ { <i>ox_{rand}</i> }.
6.	Delete the similar individuals in <i>OS</i> within the minimum distance threshold <i>Dis_eps</i> $//Dis_eps = \min(\sqrt{0.001^2n}, 5e^{-4n} \min(xu - xl))$.
7.	Calculate the diversity-based objective for each <i>x_i</i> in <i>CS</i> , i.e., $\hat{f}_{div}(x_i) = \hat{f}(x_i) / Dist(x_i)$.
8.	Calculate the diversity-based constraints for each <i>x_i</i> in <i>CS</i> , i.e., $\hat{g}_{j,div}(x_i) = \hat{g}_{j,div}(x_i) / Dist(x_i), \forall j \in \{1, \dots, t\}$.
9.	Select the top <i>N</i> ₁ − <i>N_{OS}</i> individuals from <i>CS</i> based FR by using the diversity-based objective and constraints and store them into <i>OS</i> .

Problem	Dimension	Type of objective (estimate feasibility ratio)	L(N)a
g01	13	Quadratic (0.0111%)	9(0)6
g02	20	nonlinear (99.9971%)	0(2)1
g04	5	quadratic (52.1230%)	0(6)2
g06	2	cubic (0.0066%)	0(2)2
g07	10	quadratic (0.0003%)	3(5)6
g08	2	nonlinear (0.8560%)	0(2)0
g09	7	polynomial (0.5121%)	0(4)2
g10	8	linear (0.0010%)	3(3)6
g12	3	quadratic (4.7713%)	0(1)0
g16	5	nonlinear (0.0204%)	4(34)4
g18	9	quadratic (0.0000%)	0(13)6
g19	15	nonlinear (33.4761%)	0(5)0
g24	2	linear (79.6556%)	0(2)2

Table 1. The main characteristics of thirteen benchmark problems from CEC2006. Note: a is the number of active constraints at x, and L and N are the numbers of linear inequality constraints and nonlinear inequality constraints, respectively.

Distributed Central Region Local Exploration (DCRLE)

As the algorithm progresses, the search scope tends to concentrate on a small range of more promising regions. Thus, the corresponding local regions or surrogates of many local searches may be extremely similar, since their corresponding central solutions are closer to each other, resulting in a limited exploration of new solution spaces. This limitation not only reduces the diversity of the algorithm but also increases the risk of local optima, as the algorithm may repeatedly evaluate similar solutions while neglecting other potentially high-quality solutions. As iterations continue, an excessive focus on these similar regions can lead to a decline in the algorithm’s innovative capacity, making it difficult to escape the current local optimum, thereby affecting the overall search efficiency and effectiveness. Therefore, designing effective mechanisms to expand the search range and encourage the exploration of more diverse solution areas will help enhance the algorithm’s performance and increase the likelihood of finding global optima. Therefore, as shown in **Algorithm 4**, distributed central region local exploration (DCRLE) is designed to effectively allocate uniformly distributed local searches. It utilizes affinity propagation clustering and mathematical modeling to identify uniformly distributed central solutions. However, we found that during the progression of the algorithm, two different clustering situations are shown in Figs. 5 and 6. We take g6 from CEC2006 for an example. In these two figures, each point represents an individual and the different clusters generated by APC are distinguished by points of different colors. The polygons formed by connecting points of the same color represent the areas covered by each cluster. x and y represent two different dimensions of the design variables. Figure 5 represents the clustering situation at the initial stage of the algorithm where affinity propagation clustering divides the population into four clusters, represented by the four polygons in Fig. 5. However, as the algorithm progresses, the population tends to exhibit over-clustering due to low diversity, as shown in Fig. 6. When the number of iterations reaches 573, as indicated in the lower left corner of Fig. 6, each point represents its own cluster, with only one point in each cluster. This occurs because they lack sufficient similarity or shared features to form larger clustering structures, and they do not have enough similarity with other points to form larger clusters. At this point, if we directly perform local searches on each cluster, it will lead to excessive redundancy in the search areas. Then the mathematical modeling method shown in lines 1–7 of **Algorithm 4** is used to reselect the central points for well-distributed local searches. For a detailed elaboration, the step-by-step introduction is shown below.

Step 1 in line 1, the cluster results of APC provide effective guidance to arrange the well-distributed local search.

Step 2 in lines 2–8, when the population demonstrates an excessive number of clusters resulting from high similarity among individuals, three central individuals are selected or generated based on different distributed information to ensure that the central regions of local searches designed around these individuals are diversified.

Step 3 in line 10, to determine the local search scope for each cluster, the K nearest points to each cluster center cx_k in the database DB are stored into MX_k . K is calculated as follows.

$$K = \max((D + 1)(D + 2)/2, 100) \tag{17}$$

Prob.	SGDLCO		C ³ ODE		CORCO		PROFI		DeCODE	
	Mean(Std)		Mean(Std)		Mean(Std)		Mean(Std)		Mean(Std)	
g01	-1.50E+01(8.93E-08)		NaN+		-5.22E-01(1.74E+00)+		NaN+		-1.11E+00(7.91E-01)+	
g02	-4.26E-01(5.14E-02)		-2.01E-01(1.81E-02)+		-1.55E-01(1.67E-02)+		-2.19E-01(1.31E-02)+		-1.85E-01(1.74E-02)+	
g04	-3.07E+04(2.73E-06)		-3.00E+04(1.93E+02)+		-2.96E+04(1.40E+02)+		-3.01E+04(1.01E+02)+		-3.00E+04(1.70E+02)+	
g06	-6.96E+03(1.16E-04)		-4.82E+03(1.25E+03)+		-4.68E+03(1.30E+03)+		-4.72E+03(1.09E+03)+		-3.75E+03(1.01E+03)+	
g07	2.43E+01(1.04E-05)		2.04E+03(1.65E+03)+		1.68E+02(2.58E+01)+		1.00E+03(3.35E+02)+		NaN+	
g08	-7.72E-02(3.06E-02)		-9.02E-02(5.54E-03)~		-6.00E-02(2.90E-02)~		-9.15E-02(4.11E-03)~		-8.63E-02(1.01E-02)~	
g09	6.81E+02(5.65E-02)		1.32E+03(3.42E+02)+		5.53E+04(2.65E+05)~		1.23E+03(2.20E+02)+		1.45E+03(4.45E+02)+	
g10	7.05E+03(3.44E-03)		NaN+		NaN+		NaN+		NaN+	
g12	-9.94E-01(7.96E-03)		-9.92E-01(6.44E-03)~		-1.00E+00(4.70E-05)-		-9.90E-01(5.81E-03)~		-9.94E-01(4.79E-03)~	
g16	-1.91E+00(9.02E-10)		-1.35E+00(1.87E-01)+		-1.12E+00(0.00E+00)+		-1.29E+00(1.55E-01)+		NaN+	
g18	-7.82E-01(9.68E-02)		NaN+		NaN+		NaN+		NaN+	
g19	3.33E+01(2.70E-01)		1.55E+03(6.37E+02)+		7.94E+02(5.39E+02)+		9.88E+02(2.90E+02)+		1.15E+03(3.18E+02)+	
g24	-5.51E+00(1.21E-10)		-5.38E+00(7.50E-02)+		-4.97E+00(3.50E-01)+		-5.38E+00(5.13E-02)+		-5.32E+00(1.14E-01)+	
+/-/-	Win/Tie/Loss		11/2/0		10/2/1		11/2/0		11/2/0	

Table 2. Experimental function values among SGDLCO, C³ODE, CORCO, PROFI, and DeCODE on thirteen test problems from CEC2006.

Prob.	SGDLCO	C ² ODE	CORCO	FROFI	DeCODE
	ER	ER	ER	ER	ER
g01	100%	0%	8%	0%	8%
g02	100%	100%	100%	100%	100%
g04	100%	100%	100%	100%	100%
g06	100%	100%	60%	100%	60%
g07	100%	8%	12%	8%	0%
g08	100%	100%	100%	100%	100%
g09	100%	100%	96%	100%	100%
g10	100%	0%	0%	0%	0%
g12	100%	100%	100%	100%	100%
g16	100%	60%	8%	80%	0%
g18	100%	0%	0%	0%	0%
g19	100%	100%	100%	100%	100%
g24	100%	100%	100%	100%	100%

Table 3. Effective rates among SGDLCO, C²ODE, CORCO, FROFI, and DeCODE on thirteen test problems from CEC2006.

where D represents the number of dimensions.

Step 4 in line 11, to avoid irreversibility of the Gram matrix in constructing RBF, similar individuals within the distance threshold Dis_eps are removed in each cluster. Then the local RBF models are constructed using the remaining individuals for each cluster where the compute of Dis_eps is consistent with the formula in line 6 of **Algorithm 3**.

Step 5 in lines 12–18, the inner optimization problem for conducting local search on each cluster is formulated as follows.

$$\begin{aligned}
 & \text{minimize } \hat{f}(\mathbf{x}) \\
 & \text{subject to } \hat{g}_j(\mathbf{x}) \leq 0 \quad j = 1, 2, \dots, q \\
 & \quad xL_i < x_i < xU_i, i = 1, \dots, D \\
 & \quad \hat{g}_{dis}(\mathbf{x}) \leq 0
 \end{aligned} \tag{18}$$

where $\hat{f}(\mathbf{x})$ and $\hat{g}_j(\mathbf{x})$ are the RBF surrogates built with the K nearest points for the original objective and the j -th constraint, respectively. xU_i and xL_i respectively represents the upper and lower bounds of i -th dimension in the current local search scope. $\hat{g}_{dis}(\mathbf{x})$ donates the distance constraint calculated based on the diversity threshold which is computed in Eq. (19).

$$\hat{g}_{dis}(\mathbf{x}_i) = Dis_eps - Dist(\mathbf{x}_i) \tag{19}$$

Step 6 in lines 19–21, the current population in inner optimization is regenerated when the current population trapped in a local infeasible region, i.e., all the individuals are infeasible and their constraint violations show extremely small differences.

Step 7 in lines 22–33, based on the recommendation in CoDE⁶⁶, three different combinations of mutation operations and parameters are employed to generate diverse offspring solutions.

Step 8 in line 36, the best solution lv_k is obtained by completing the inner optimization for each central solution.

Prob.	SGDLCO	C ² ODE	CORCO	PROFI	DeCODE
D = 10	Mean(Sth)	Mean(Sth)	Mean(Sth)	Mean(Sth)	Mean(Sth)
C01	-4.96E-01(9.48E-02)	-3.20E-01(4.00E-02)+	1.75E-01(2.60E-02)+	-3.52E-01(4.11E-02)+	-2.87E-01(1.53E-02)+
C07	2.33E+03(3.12E+03)	2.12E+09(1.06E+09)+	3.99E+09(3.49E+09)+	2.20E+09(1.16E+09)+	3.02E+09(7.86E+08)+
C08	8.98E+03(9.42E+03)	2.68E+09(1.65E+09)+	1.18E+10(9.29E+09)+	2.59E+09(1.45E+09)+	2.74E+09(8.64E+08)+
C13	-5.59E+01(5.57E+00)	-1.13E+01(1.69E+01)+	NaN+	NaN+	NaN+
C14	2.93E+11(8.97E+11)	6.90E+13(4.56E+13)+	1.71E+13(1.00E+13)+	9.90E+13(4.46E+13)+	8.88E+13(6.51E+13)+
C15	2.02E+14(0.00E+00)	2.27E+14(9.24E+13)-	NaN+	1.96E+14(1.33E+14)-	NaN+
+/-/-	Win/Tie/Loss	5/1/0	6/0/0	5/1/0	6/0/0

Table 4. Experimental function values among SGDLCO, C²ODE, CORCO, PROFi, and DeCODE on six 10-D test problems from CEC2010.

	SGDLCO	C ² ODE	CORCO	FROFI	DeCODE
Prob.	ER	ER	ER	ER	ER
C01	100%	100%	100%	100%	100%
C07	100%	100%	100%	100%	100%
C08	100%	100%	100%	100%	100%
C13	100%	12%	0%	0%	0%
C14	100%	100%	100%	100%	92%
C15	4%	16%	0%	8%	0%

Table 5. Effective rates among SGDLCO, C²ODE, CORCO, FROFI, and DeCODE on six 10-D test problems from CEC2010.

Algorithm 4 Distributed Central Region Local Exploration (DCRLE)	
Input:	Pop, DB, NP , total evaluation times: $totalFES, K, Dis_eps$.
Output:	LV
1.	The Pop is clustered into np clusters, such as $CX_1, \dots, CX_{np}$ based on APC.
2.	If $np \geq [0.2NP]$ // Re-select three individuals based on the distribution of Pop when the clustered number is large.
3.	Reset $np = 3$, for each individual in Pop , calculate the minimum distance $Dist(x_i) = \min_{x_i \in Pop} \ x_i - DB\ $.
4.	Select an individual such as cx_1 and cx_2 with the maximum and minimum i.e., $Dist(cx_1)$ and $Dist(cx_2)$.
5.	Obtain the mean individual of Pop , i.e., $cx_{3,j} = \sum_{i=1}^{NP} x_{i,j} / NP, \forall j \in \{1, \dots, n\} \forall x_i \in Pop$.
6.	Else If
7.	Obtain the mean individual of CX_k for each dimension, i.e., $cx_{3,j} = \sum_{i=1}^{NP} x_{i,j} / NP, \forall j \in \{1, \dots, n\} \forall x_i \in CX_k$.
8.	End If
9.	For $k = 1 : np$
10.	Store the K nearest individuals to cx_k into MX_k from DB and $MX_k = MX_k \cup NP_k$.
11.	Delete the similar individuals in MX_k and construct local RBF models, i.e., $\hat{f}^k, \hat{g}_1^k, \dots, \hat{g}_t^k$, for objective and constraints by using the remaining individuals in MX_k .
12.	While $FES \leq totalFES$
13.	Initialize the population TX .
14.	For each tx_i in TX
15.	Calculate the minimum distance $Dist(tx_i) = \min_{tx_i \in TX} \ tx_i - DB\ $.
16.	Predict the objective and constraints for tx_i by local RBF models.
17.	Evaluate the distance constraint for tx_i , i.e., $g_{dis} = Dis_eps - Dist(tx_i)$.
18.	End For
19.	If all the individuals are infeasible and their constraint violations show extremely small differences
20.	Regenerate the population such as TX , and predict them by local RBF models.
21.	End If
22.	For each tx_i in TX
23.	If $rand(0, 1) \leq 1/3$
24.	$tv_i = tx_i + F(tx_{best} - tx_i) + F(tx_{r1} - tx_{r2})$, tx_{r1} and tx_{r2} are two distinct individuals selected from TX , and $F = 0.6$.
25.	Else If $rand(0, 1) > 2/3$
26.	$tv_i = tx_i + F(tx_{r1} - tx_i)$, tx_{r1} is a distinct individual selected from TX , and $F = 0.8$.
27.	Else
28.	$tv_i = tx_i + F(tx_{r1} - tx_2)$, tx_{r1} and tx_{r2} are two distinct individuals selected from TX , and $F = 1.0$.
29.	End If
30.	Obtain uv_i by conducting binomial crossover on tv_i and tx_i .
31.	If uv_i is better than tx_i based on FR.
32.	$tx_i = uv_i$.
33.	End If
34.	End For
35.	End While
36.	Store the best one lv_k from TX , and archive lv_k into $LV = \{lv_1, \dots, lv_{np}\}$.
37.	End For

Prob.	SGDLCO	C ² ODE	CORCO	PROFI	DeCODE
D=30	Mean(Sth)	Mean(Sth)	Mean(Sth)	Mean(Sth)	Mean(Sth)
C01	-3.86E-01(4.49E-02)	-2.19E-01(1.41E-02)+	4.46E+14(1.80E+14)+	-1.90E-01(1.05E-02)+	-2.38E-01(2.23E-02)+
C07	2.67E+07(1.40E+07)	7.98E+10(7.30E+09)+	3.64E+10(1.24E+10)+	5.62E+10(1.50E+10)+	7.28E+10(1.32E+10)+
C08	1.93E+09(1.05E+09)	6.83E+10(1.47E+10)+	8.64E+10(3.70E+10)+	6.21E+10(1.47E+10)+	5.74E+10(1.09E+10)+
C13	-4.78E+01(7.85E+00)	NaN+	NaN+	NaN+	NaN+
C14	1.68E+14(5.71E+13)	3.41E+14(1.11E+14)+	1.88E+14(1.83+14)~	3.16E+14(7.17E+13)+	4.22E+14(1.29E+14)+
C15	5.21E+14(2.41E+14)	7.47E+14(1.97E+14)+	4.46E+14(1.80E+14)~	4.61E+14(1.47E+14)~	NaN+
+/-/-	Win/Tie/Loss	6/0/0	4/2/0	5/1/0	6/0/0

Table 6. Experimental function values among SGDLCO, C²ODE, CORCO, PROFi, and DeCODE on six 30-D test problems from CEC2010.

Prob.	SGDLCO	C ² ODE	CORCO	FROFI	DeCODE
	ER	ER	ER	ER	ER
C01	100%	100%	40%	100%	100%
C07	100%	100%	100%	100%	100%
C08	100%	100%	100%	100%	100%
C13	100%	0%	0%	0%	0%
C14	100%	100%	100%	100%	92%
C15	28%	16%	40%	20%	0%

Table 7. Effective rates among SGDLCO, C²ODE, CORCO, FROFI, and DeCODE on six 30-D test problems from CEC2010.

Experimental studies
Experimental settings

The performance of the proposed algorithm SGDLCO is verified by using thirteen widely used inequality-constrained problems from CEC2006, as well as 10-dimensional and 30-dimensional inequality-constrained problems from CEC2010 and CEC2017. The main characteristics of these thirteen benchmark problems from CEC2006 are listed in Table 1. And the main characteristics of other test problems from CEC2017 are listed in Table A1 in the Appendix. The maximum number of fitness evaluations *MaxFEs* is set to 1000 for all the test problems. All experimental results are obtained over 25 independent runs in Matlab R2023a. To systematically evaluate and compare the performance of SGDLCO with other different algorithms in solving specific optimization problems, the effective rate (ER) metric shown in Eq. (19) is used to measure the ratio of effective evaluations to total evaluations. If at least one feasible solution is found during one evaluation, then that evaluation is considered effective. In Eq. (20), *N_{effective}* and *N_{max}* respectively represent the number of effective evaluations and the total number of evaluations. Additionally, in the following table, “Mean” and “Std” respectively represent the average and standard deviation of the objective function values obtained among all effective runs.

$$ER = \frac{N_{effective}}{N_{max}} \tag{20}$$

Comparison with four evolutionary algorithms

Four excellent evolutionary algorithms, i.e., C²oDE⁶⁷, CORCO⁶⁸and FROFI⁶⁹, DeCODE⁷⁰ are selected for comparison with SGDLCO. C²oDE utilizes three different trial vector generation strategies to balance diversity and convergence, while achieving a balance between constraints and the objective function. It performs better or at least comparably to other state-of-the-art methods on multiple benchmark test functions. CORCO proposes a new constrained optimization evolutionary algorithm which utilizes the correlation between constraints and the objective function for the first time, and it balances this correlation through a correlation index. FROFI introduces a novel replacement mechanism and mutation strategy, and effectively utilizes objective function information to alleviate the excessive bias of known feasibility rules, thereby enhancing its robustness. DeCODE utilizes the decomposition-based multi-objective optimization method to solve constrained optimization problems by transforming them into bi-objective optimization problems and decomposing them into scalar optimization subproblems.

The experimental results (objective function values) of all these algorithms on the 13 test problems from CEC2006 are listed in Table 2. The effective rate of all these algorithms on the 13 test problems from CEC2006 are listed in Table 3. Tables 4 and 5 list the test results of each algorithm on six 10-dimensional and 30-dimensional problems in CEC2010, respectively, while Tables 6 and 7 list the test results of each algorithm on seven 10-dimensional and five 30-dimensional problems in CEC2017, respectively. In these tables, t-test are employed, as shown in Table 3, where “Win” indicates the number of test problems where SGDLCO outperforms the algorithm, “Tie” indicates the number of test problems where SGDLCO performs comparably to the algorithm, and “Loss” indicates the number of test problems where SGDLCO performs worse than the algorithm.

In Table 2, on nine test problems (g1, g2, g4, g6, g7, g10, g16, g18, g19, g24), SGDLCO is able to achieve solutions that are significantly better than those of the other four algorithms. This indicates that within 1000 evaluation attempts, SGDLCO is able to find optimal solutions that are significantly better than those of the other four algorithms. In Table 3, for eight test problems (i.e., g01, g07, g10, g16, g18) with smaller feasibility ratio, SGDLCO is able to achieve a 100% ER on these test problems. This means that, compared to these four algorithms, SGDLCO is always able to find at least one feasible solution when dealing with these test problems.

As shown in Table 4, SGDLCO performs excellently on the majority of problems, outperforming other algorithms on the C01, C07, C08, C13, and C14 problems. As shown in Tables 5 and 7, for C13 problem with a low feasibility rate, SGDLCO consistently finds feasible solutions in every run, whereas other algorithms fail to obtain feasible solutions. As shown in Table 6, SGDLCO achieved better or comparable results than other algorithms on all problems. As shown in Table 8, SGDLCO performs excellently on the C01, C02, C04, C05, C13, and C22 problems, outperforming other algorithms on these problems. As shown in Table 9, for C13 and C22 problems with low feasibility rates, SGDLCO consistently finds feasible solutions in every run, whereas other algorithms fail to obtain a feasible solution. Similar results can also be seen in Tables 10 and 11. Therefore,

Prob.	SGDLCO	C ² ODE	CORCO	PROFI	DeCODE
D = 10	Mean(Sth)	Mean(Sth)	Mean(Sth)	Mean(Sth)	Mean(Sth)
C01	5.61E-01(1.43E+00)	3.89E+03(1.07E+03)+	1.09E+04(3.36E+03)+	4.52E+03(1.30E+03)+	5.60E+03(9.07E+02)+
C02	2.50E+00(6.72E+00)	3.83E+03(1.14E+03)+	3.69E+03(1.01E+03)+	4.00E+03(1.22E+03)+	3.84E+03(1.62E+03)+
C04	5.37E+01(1.27E+01)	1.50E+02(1.88E+01)+	1.52E+02(1.38E+01)+	1.53E+02(1.95E+01)+	1.53E+02(2.11E+01)+
C05	7.40E+00(1.22E+00)	1.64E+04(1.51E+04)+	5.73E+03(4.80E+03)+	6.34E+04(6.04E+04)+	1.84E+04(8.58E+03)+
C13	3.89E+01(5.00E+01)	NaN+	NaN+	NaN+	NaN+
C20	2.41E+00(4.47E-01)	2.46E+00(3.78E-01)~	2.29E+00(4.46E-01)~	4.48E+00(8.05E-01)+	2.33E+00(1.54E-01)~
C22	8.89E+01(5.07E+01)	NaN+	NaN+	NaN+	NaN+
+/-/-	Win/Tie/Loss	6/1/0	6/1/0	7/0/0	6/1/0

Table 8. Experimental function values among SGDLCO, C²ODE, CORCO, PROFI, and DeCODE on seven 10-D test problems from CEC2017.

Prob.	SGDLCO	C ² ODE	CORCO	FROFI	DeCODE
	ER	ER	ER	ER	ER
C01	100%	100%	100%	100%	100%
C02	100%	100%	100%	100%	100%
C04	100%	100%	100%	100%	100%
C05	100%	100%	100%	32%	100%
C13	100%	0%	0%	0%	0%
C20	100%	100%	100%	100%	100%
C22	100%	0%	0%	0%	0%

Table 9. Effective rates among SGDLCO, C²ODE, CORCO, FROFI, and DeCODE on six 10-D test problems from CEC2017.

SGDLCO significantly outperforms the other four algorithms across various test problems, demonstrating its superiority and stability in handling inequality-constrained optimization problems.

Comparison with four SAEAs

Many SAEAs are developed to solve expensive constrained optimization problems. Seven out of these algorithms, i.e., GLoSADE⁵, SADE-CV_{SR}³⁷, SADE-CV_{FR}³⁷, SParEA⁷¹, MPMLS³⁵, SA-TSDE⁷² and SACCDE⁴⁰ provide competitive performance than others on solving classical test suites. Therefore, they are employed to compare with SGDLCO on three widely used test suites, i.e., IEEE CEC2006⁶¹, IEEE CEC2010⁶² and IEEE CEC2017⁶³. To be consistent with the three comparison methods, we suppose that both the objective and constraints are simultaneously evaluated by running one expensive simulation. The comparison results of these algorithms on benchmark problems from IEEE CEC2006, IEEE CEC2010 and IEEE CEC2017 are respectively shown in Tables 12, 13 and 14.

In Table 12, SGDLCO is capable of obtaining near-optimal solutions for the majority of relatively simple problems in IEEE CEC2006 and outperforms these comparison algorithms on most functions. This indicates that SGDLCO has the ability to quickly locate the global optimum for simple problems. Furthermore, in Table 13, for relatively complex problems such as those in IEEE CEC2010, SGDLCO also significantly outperforms these comparison algorithms on most functions, which indicates that SGDLCO has a significant performance advantage on CEC2010 test suites. In Table 14, SGDLCO has a significant advantage over only four of the seven algorithms such as GLoSADE, SADE-CV_{SR}, SADE-CV_{FR}, and MPMLS, and SGDLCO has comparable performance with the other three algorithms. This indicates that the performance of SGDLCO varies across different test suites and the performance of SGDLCO on CEC2017 does not seem to be significant. In order to analyze the overall performance of the SGDLCO algorithm on the three test suites more clearly, the statistical results of the overall performance of the algorithm against the three test sets are presented in Sect.4.4. In addition, the detailed comparative analysis results with each algorithm are as follows:

- (1) For GLoSADE, in CEC2006, SGDLCO achieves better results on seven out of thirteen problems. In CEC2010, SGDLCO outperforms GLoSADE on six problems and achieves comparable results on two problems. Additionally, in CEC 2017, SGDLCO achieves better results on ten out of twelve problems.
- (2) For SADE-CV_{SR} and SADE-CV_{FR}, in CEC2006, SGDLCO achieves better results on eleven out of thirteen problems. In CEC2010, SGDLCO outperforms both SADE-CV_{SR} and SADE-CV_{FR} on ten problems. Additionally, in CEC 2017, SGDLCO achieves better results on eleven out of twelve problems compared to both algorithms.
- (3) For SParEA, in CEC2006, SGDLCO achieves better results on six out of thirteen problems and had similar results on five problems. In CEC2010, SGDLCO outperforms SParEA on six problems. Additionally, in CEC 2017, SGDLCO achieves better results on three out of twelve problems and had similar results on seven problems with SParEA. It is worth noting that while the t-test results were similar on some problems, such as C22 in Table 15 and C04 in the six row of Table 14, SGDLCO consistently obtained significantly better average values for these specific problems compared to SParEA.
- (4) For MPMLS, in CEC2006, SGDLCO achieves better results on seven out of thirteen problems. In CEC2010, the performance of SGDLCO is significantly better than MPMLS on five problems, and SGDLCO obtains comparable results than MPMLS on five problems. In CEC2017, SGDLCO outperforms MPMLS on six problems, and SGDLCO obtains comparable results than MPMLS on four problems.
- (5) For SA-TSDE, in CEC2006, SGDLCO achieves better results on eight out of thirteen problems and obtains similar results on four problems. In CEC2010, the performance of SGDLCO is significantly better than SA-TSDE on six problems and SGDLCO obtains comparable results than SA-TSDE on four problems. In the twelve problems tested in CEC2017, SGDLCO outperforms SA-TSDE on four problems, and SGDLCO obtains comparable results on another four.
- (6) For SACCDE, in CEC2006, SGDLCO achieves better results on five out of thirteen problems, and SGDLCO has similar results than SACCDE on five problems. In the twelve problems tested in CEC2010, SGDLCO achieves better results on six problems, and SGDLCO obtains comparable results on six problems. In CEC2017, SGDLCO outperforms SACCDE on three problems, and SGDLCO obtains comparable results

Prob.	SGDLCO	C ² ODE	CORCO	FROFI	DeCODE
D = 30	Mean(Sth)	Mean(Sth)	Mean(Sth)	Mean(Sth)	Mean(Sth)
C01	3.38E+04(8.24E+03)	6.27E+04(6.71E+03)+	8.68E+04(1.55E+04)+	5.54E+04(8.76E+03)+	7.79E+04(2.00E+04)+
C02	1.50E+04(3.37E+03)	2.71E+05(2.38E+05)+	1.03E+05(3.65E+05)~	5.21E+05(3.06E+05)+	4.28E+05(3.37E+05)+
C04	3.69E+02(2.16E+01)	8.31E+02(8.31E+01)+	7.13E+02(1.75E+01)+	7.49E+02(5.85E+01)+	8.32E+02(7.89E+01)+
C05	4.18E+02(7.82E+01)	NaN+	NaN+	NaN+	NaN+
C20	1.05E+01(6.90E-01)	1.04E+01(7.54E-01)~	1.09E+01(5.30E-01)~	1.32E+01(1.37E+00)+	1.04E+01(4.88E-01)~
+/-/-	Win/Tie/Loss	4/1/0	3/2/0	5/0/0	4/1/0

Table 10. Experimental function values between SGDLCO, C²ODE, CORCO, FROFI, and DeCODE on five 30-D test problems from CEC2017.

Prob.	SGDLCO	C ² ODE	CORCO	FROFI	DeCODE
	ER	ER	ER	ER	ER
C01	100%	100%	100%	100%	100%
C02	100%	32%	24%	8%	16%
C04	100%	100%	100%	100%	100%
C05	100%	0%	0%	0%	0%
C20	100%	100%	100%	100%	100%

Table 11. Effective rates among SGDLCO, C²ODE, CORCO, FROFI, and DeCODE on five 30-D test problems from CEC2017.

than SACCDE on six problems. This indicates that in the three test suites, SGDLCO outperforms or performs comparably to SACCDE in most problems.

Wilcoxon signed rank test for comparison

To further compare the overall performance of SGDLCO with these classical algorithms on the aforementioned test problems, we employed the Wilcoxon signed rank test⁷³ to comprehensively assess their performance differences. All Wilcoxon signed rank test results are listed in Table 15. Here, different dimensions of the same function (e.g., 10D C01 and 30D C01) are considered as distinct problems.

In Table 15, it can be observed that SGDLCO shows a significant improvement over SADE-CV_{SR}, SADE-CV_{FR} with a level of significance level $\alpha = 0.01$, over GLoSADE with $\alpha = 0.02$, over SA-TSDE with $\alpha = 0.05$, over SACCDE with $\alpha = 0.10$, over SPArEA with $\alpha = 0.20$, and over MPMLS with $\alpha = 0.25$.

Effectiveness of some strategies in SGDLCO

In this subsection, five 10-D problems in CEC2017 are chosen to discuss the effectiveness of several strategies proposed in this paper. The maximum number of fitness evaluations *MaxFEs* is set to 1000, and the rest parameters about SGDLCO are set the same as that suggested in subsection 4.1.

(1) Effectiveness of classification cooperation mutation operation

To verify the effectiveness of the classification cooperation mutation operation (CCMO) proposed in SGDLCO, a variant called SGDLCO_noCCMO is introduced for detailed comparison. SGDLCO_noCCMO replaces the classification cooperation mutation operation (CCMO) in SGDLCO with the DE/best/2 method as shown in Eq⁸.

Figure 7 lists the comparison results of SGDLCO and SGDLCO_noCCMO on the five 10-D test problems from CEC2017. From Fig. 7, it can be seen that SGDLCO achieved better or comparable results to SGDLCO_noCCMO except for C05. This demonstrates that CCMO effectively utilizes information from all individuals to obtain more accurate solutions. Figure 8 lists the standard deviations of the operational results of SGDLCO and SGDLCO_noCCMO on the five 10-dimensional test problems from CEC2017. From the table, we can see that SGDLCO exhibits more stable results than SGDLCO_noCCMO on most problems.

(2) Effectiveness of three-layer adaptive screening

To maintain population diversity during the selection process and avoid the population getting trapped in local optima, this study constructs a variant of SGDLCO called SGDLCO_noTLAS. SGDLCO_noTLAS replaces TLAS in SGDLCO with FR.

Figure 9 presents the experimental results of SGDLCO and SGDLCO_noTLAS on five selected problems from CEC2017. From Fig. 9, we can see that SGDLCO achieves better results than SGDLCO_noTLAS. These experimental results indicate that TLAS effectively maintains the diversity of the population, enabling effective exploration of the solution space and enhancing the quality of individuals within the population. Therefore, TLAS significantly increases the chances of finding the global optimum in the solution space, rather than getting stuck in local optima. Figure 10 lists the standard deviations of the operational results of SGDLCO and SGDLCO_noTLAS on the five 10-dimensional test problems from CEC2017. From the table, we can see that SGDLCO exhibits more stable results than SGDLCO_noTLAS in 3 out of the 5 problems.

(3) Effectiveness of distributed central region local exploration

To verify the effectiveness of the DCRLE proposed in this paper, a variant of SGDLCO, called SGDLCO_noDCRLE, was introduced. SGDLCO_noDCRLE replaces the DCRLE in SGDLCO with the local surrogate-assisted search phase detailed introduced in GLoSADE. The core difference between the local surrogate-assisted search phase in GLoSADE and the DCRLE is that the first approach performs complete local search processes for each population individual, while the DCRLE only performs local searches for selected individuals with well-distribution and potentiality.

Figure 11 lists the test results of SGDLCO and SGDLCO_noDCRLE on five problems from CEC2017. It can be seen that SGDLCO achieved better results on 4 out of 5 tested problems, with C20 achieving comparable results between the two. And from Fig. 12 we can observe that SGDLCO achieves more stable results than SGDLCO-noDCRLE on all problems except C20. This indicates that SGDLCO has a stronger search capability compared to SGDLCO_noDCRLE. This suggests that DCRLE can effectively allocate uniformly distributed local

Prob.	SGDLCO		GloSADE		SADE-CV _{sr}		SADE-CV _{pr}		SParEA		MPMLS		SA-TSDE		SACDE	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
g01	-1.50E+01	(8.93E-08)	-1.50E+01	(8.20E-07)~	NaN+		NaN+		-1.50E+01	(9.06E-15)~	-1.50E+01	(9.46E-03)+	-1.50E+01	(9.59E-02)~	-1.50E+01	(2.22E-13)~
g02	-4.26E-01	(5.14E-02)	-3.03E-01	(2.65E-02)+	-2.32E-01	(3.58E-02)+	-2.14E-01	(2.53E-02)+	-2.64E-01	(3.33E-02)+	-3.99E-01	(5.91E-02)~	-3.19E-01	(6.09E-02)+	-4.67E-01	(4.72E-01)~
g04	-3.07E+04	(2.73E-06)	-3.06E+04	(2.02E-06)+	-3.04E+04	(1.58E+02)+	-3.03E+04	(2.97E+02)+	-3.06E+04	(7.43E-13)+	-3.07E+04	(1.39E-03)+	-3.07E+04	(4.09E-03)+	-3.07E+04	(2.72E-06)+
g06	-6.96E+03	(1.16E-04)	-6.96E+03	(4.44E-01)~	-6.87E+03	(2.08E+02)~	-5.54E+03	(1.71E+03)+	-6.96E+03	(3.70E-11)~	-6.59E+03	(9.40E+02)~	-6.96E+03	(9.37E-05)-	-6.96E+03	(4.43E-05)-
g07	2.43E+01	(1.04E-05)	2.43E+01	(1.06E-06)~	NaN+		NaN+		2.43E+01	(5.53E-12)~	2.45E+01	(1.57E-01)+	2.59E+01	(2.14E+00)+	2.43E+01	(4.20E-04)+
g08	-7.72E-02	(3.06E-02)	-9.58E-02	(3.34E-10)-	-8.76E-02	(2.57E-02)~	-3.96E-02	(4.57E-02)+	-9.27E-02	(1.33E-02)~	-9.58E-02	(1.40E-15)-	-7.44E-02	(3.19E-02)~	-9.58E-02	(1.47E-09)-
g09	6.81E+02	(5.65E-02)	9.15E+02	(1.09E+02)+	2.01E+03	(1.12E+03)+	1.96E+04	(4.64E+04)~	6.95E+02	(1.61E+01)+	6.81E+02	(1.42E-01)-	7.00E+02	(1.83E+01)+	7.45E+02	(1.12E+02)+
g10	7.05E+03	(3.44E-03)	7.08E+03	(4.04E+01)+	NaN+		NaN+		7.05E+03	(1.44E-07)~	7.10E+03	(3.12E+01)+	7.24E+03	(2.08E+02)+	7.05E+03	(1.22E+00)-
g12	-9.94E-01	(7.96E-03)	-9.98E-01	(2.34E-03)-	-9.16E-01	(2.25E-02)+	-9.37E-01	(3.58E-02)+	-1.00E+00	(4.57E-10)-	-1.00E+00	(0.00E+00)-	-9.90E-01	(1.45E-02)~	-9.99E-01	(1.19E-02)~
g16	-1.91E+00	(9.02E-10)	-1.85E+00	(6.46E-02)+	-1.13E+00	(1.74E-01)+	-1.42E+00	(1.41E-01)+	-1.90E+00	(5.44E-04)+	-1.91E+00	(2.69E-05)+	-1.84E+00	(1.02E-01)+	-1.91E+00	(8.65E-09)+
g18	-7.82E-01	(9.68E-02)	-7.68E-01	(8.63E-02)~	NaN+		NaN+		-8.58E-01	(3.82E-02)-	-7.66E-01	(9.52E-02)~	-7.84E-01	(9.34E-02)~	-8.47E-01	(1.91E-01)~
g19	3.33E+01	(2.70E-01)	3.82E+01	(7.02E+00)+	1.14E+03	(2.80E+02)+	1.14E+03	(4.42E+02)+	8.63E+01	(6.30E+01)+	3.83E+01	(2.85E+00)+	9.05E+01	(3.02E+01)+	3.49E+01	(9.15E+00)~
g24	-5.51E+00	(1.21E-10)	-5.50E+00	(5.45E-08)+	-5.13E+00	(4.83E-01)+	-5.38E+00	(4.00E-01)~	-5.50E+00	(1.06E-08)+	-5.51E+00	(1.43E-12)+	-5.51E+00	(9.18E-10)+	-5.51E+00	(1.55E-05)+
+/-/-	Win/Tie/Loss		7/4/2		11/2/0		11/2/0		6/5/2		7/3/3		8/4/1		5/5/3	

Table 12. Comparison results (function values) among the compared eight algorithms on test problems from IEEE CEC 2006.

Prob.	D	SGD LCO	GLoSADe		SADE-CV _{SR}		SADE-CV _{FR}		SParEA	MPMLS		SA-TSDE		SAC CDE
		Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	
C01	10	-4.96E-01(9.48E-02)	-3.73E-01(5.95E-02)+	-3.37E-01(5.54E-02)+	-3.58E-01(6.47E-02)+	-3.28E-01(9.59E-02)+	-4.35E-01 (5.36E-02)+	-5.22E-01(7.68E-02)~	-3.60E-01(9.25E-02)+					
	30	-3.86E-01(4.49E-02)	-2.23E-01(2.11E-02)+	-1.65E-01(1.53E-02)+	-1.68E-01(1.24E-02)+	-3.07E-01(3.96E-02)+	-3.42E-01(5.44E-02)+	-2.87E-01(4.35E-02)+	-3.88E-01(-2.88E-01)~					
C07	10	2.33E+03(3.12E+03)	3.99E+05(2.69E+05)+	5.95E+09(3.02E+09)+	6.12E+09(3.10E+09)+	2.86E+04(3.06E+04)+	1.27E+03(2.54E+03)~	2.20E+04(2.80E+04)+	2.93E+05(4.24E+05)+					
	30	2.67E+07(1.40E+07)	5.04E+07(3.70E+07)+	9.19E+10(1.90E+10)+	9.85E+10(1.91E+10)+	7.29E+08(1.07E+09)+	2.75E+08(1.30E+09)~	4.31E+06(5.15E+06)-	1.76E+07(2.75E+07)~					
C08	10	8.98E+03(9.42E+03)	2.09E+06(1.65E+06)+	6.50E+09(3.37E+09)+	7.10E+09(3.13E+09)+	6.27E+07(6.68E+07)+	2.43E+04(2.87E+04)+	1.11E+06(1.60E+06)+	7.39E+05(7.42E+05)+					
	30	1.93E+09(1.05E+09)	1.36E+08(4.95E+07)-	9.70E+10(1.31E+10)+	9.80E+10(1.55E+10)+	3.05E+09(2.66E+09)~	3.50E+08(8.59E+08)-	1.65E+08(9.29E+07)-	3.86E+09(6.61E+09)~					
C13	10	-5.59E+01(5.57E+00)	-5.26E+01(5.52E+00)~	NaN+	NaN+	-6.00E+01(4.35E+00)-	-5.79E+01(4.48E+00)~	-5.56E+01(6.02E+00)~	-5.73E+01(4.79E+00)~					
	30	-4.78E+01(7.85E+00)	-3.36E+01(9.26E+00)+	NaN+	NaN+	-4.05E+01(2.29E+00)+	-4.29E+01(5.45E+00)+	-5.00E+01(4.26E+00)~	-5.62E+01(5.09E+01)~					
C14	10	2.93E+11(8.97E+11)	1.16E+11(1.89E+11)~	1.07E+14(5.23E+13)+	1.18E+14(5.85E+13)+	3.34E+11(5.24E+11)~	2.16E+12(3.64E+12)+	6.01E+13(4.80E+13)+	6.15E+13(4.51E+13)+					
	30	1.68E+14(5.71E+13)	5.22E+11(5.96E+11)-	3.93E+14(9.08E+13)+	4.17E+14(1.13E+14)+	1.85E+12(1.73E+12)-	1.17E+13(2.60E+13)-	2.27E+14(1.07E+14)+	3.27E+14(7.28E+13)+					
C15	10	2.02E+14(0.00E+00)	3.91E+12(6.23E+12)-	1.98E+14(9.78E+13)~	1.78E+14(6.36E+13)~	1.98E+12(2.38E+12)-	1.46E+14(1.38E+14)~	2.58E+14(2.13E+14)~	1.75E+14(2.62E+13)+					
	30	5.21E+14(2.41E+14)	6.94E+13(2.19E+14)-	6.20E+14(4.96E+13)~	5.11E+14(1.46E+14)~	1.05E+13(1.27E+13)-	3.87E+14(2.48E+14)~	1.53E+15(9.73E+14)+	5.55E+14(6.96E+14)~					
+/-/-		Win/Tie/Loss	6/2/4	10/2/0	10/2/0	6/2/4	5/5/2	6/4/2	6/6/0					

Table 13. Comparison results (function values) among the compared eight algorithms on six 10-D and 30-D test problems from IEEE CEC 2010.

Prob.	D	SGDLCO	GLoSADE	SADE-CV _{sr}		SADE-CV _{fr}		SParEA	MPMLS		SA-TSDE		SACCODE
		Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)
C01	10	5.61E-01 (1.43E+00)	2.94E+03(8.21E+02)+	6.62E+03(2.15E+03)+	6.82E+03(2.04E+03)+	5.99E-08(3.97E-08)~	1.84E+03(7.80E+02)+	4.35E+01(8.39E+01)+	4.35E+01(8.39E+01)+	8.21E+00(1.68E+01)~			
	30	3.38E+04(8.24E+03)	4.10E+04(6.16E+03)+	8.43E+04(1.25E+04)+	8.10E+04(1.26E+04)+	6.82E-03 (1.19E-03)-	4.88E+04(9.28E+03)+	1.96E+04(7.23E+03)-	1.96E+04(7.23E+03)-	1.20E+04(1.93E+04)-			
C02	10	2.50E+00(6.72E+00)	1.59E+03(4.86E+02)+	1.56E+04(2.04E+04)+	1.99E+04(2.14E+04)+	5.39E-08 (2.52E-08)~	6.00E+02(2.37E+02)+	5.95E+01(1.25E+02)~	5.95E+01(1.25E+02)~	4.15E+00(7.87E+00)~			
	30	1.50E+04(3.37E+03)	3.15E+04(7.64E+03)+	NaN+	NaN+	1.00E+04(2.50E+04)~	1.22E+04(2.95E+03)-	7.07E+03(2.67E+03)-	7.07E+03(2.67E+03)-	6.44E+03 (9.66E+03)-			
C04	10	5.37E+01(1.27E+01)	1.08E+02(1.85E+01)+	1.86E+02(3.37E+01)+	1.81E+02(3.27E+01)+	5.03E+01 (1.05E+01)~	9.03E+01(2.05E+01)+	6.56E+01(1.21E+01)+	6.56E+01(1.21E+01)+	6.01E+01(2.02E+01)~			
	30	3.69E+02(2.16E+01)	4.28E+02(3.60E+01)+	9.01E+02(7.60E+01)+	9.17E+02(8.06E+01)+	4.00E+02(6.60E+01)~	3.66E+02(2.46E+01)~	2.11E+02(3.68E+01)-	2.11E+02(3.68E+01)-	1.73E+02 (2.58E+02)-			
C05	10	7.40E+00 (1.22E+00)	4.43E+01(2.09E+01)+	3.91E+05(4.31E+05)+	1.27E+05(6.05E+04)+	2.80E+01(2.62E+01)+	7.55E+00(4.20E-01)~	8.63E+00(2.37E+00)~	8.63E+00(2.37E+00)~	5.40E+01(4.80E+01)+			
	30	4.18E+02(7.82E+01)	1.16E+04(3.48E+03)+	NaN+	NaN+	2.72E+04(1.87E+04)+	2.55E+02(2.13E+02)-	2.01E+02 (1.08E+02)-	2.01E+02(1.08E+02)-	1.14E+03(1.73E+03)~			
C13	10	3.89E+01 (5.00E+01)	1.93E+02(1.07E+02)+	NaN+	NaN+	1.14E+02(9.27E+01)+	1.23E+02(1.20E+02)+	8.11E+02(1.09E+03)+	8.11E+02(1.09E+03)+	4.19E+02(3.81E+02)+			
C20	10	2.41E+00(4.47E-01)	2.56E+00(3.51E-01)~	2.58E+00(3.22E-01)+	2.73E+00(3.22E-01)+	2.06E+00 (3.39E-01)-	2.43E+00(3.81E-01)~	2.34E+00(4.11E-01)~	2.34E+00(4.11E-01)~	2.39E+00(5.14E-01)~			
	30	1.05E+01(6.90E-01)	1.03E+01(8.64E-01)~	1.11E+01(5.64E-01)+	1.07E+01(6.83E-01)~	1.01E+01 (5.53E-01)~	1.03E+01(7.67E-01)~	1.08E+01(6.76E-01)~	1.08E+01(6.76E-01)~	1.03E+01(1.10E+01)~			
C22	10	8.89E+01 (5.07E+01)	1.20E+03(1.21E+03)+	NaN+	NaN+	1.03E+02(4.40E+01)~	6.47E+03(5.54E+03)+	2.06E+04(3.34E+04)+	2.06E+04(3.34E+04)+	1.56E+04(3.29E+04)+			
+/-/-		Win/Tie/Loss	10/2/0	11/1/0	11/1/0	3/7/2	6/4/2	4/4/4	3/6/3				

Table 14. Comparison results (function values) among the compared eight algorithms on seven 10-D and five 30-D test problems from IEEE CEC 2017.

Comparison	Num	P-value
SGDLCO vs. GLoSADE	37	1.77E-02
SGDLCO vs. SADE-CV _{SR}	27	6.74E-06
SGDLCO vs. SADE-CV _{FR}	27	1.90E-04
SGDLCO vs. SParEA	37	1.60E-01
SGDLCO vs. MPMLS	37	2.07E-01
SGDLCO vs. SACCDE	37	7.90E-02
SGDLCO vs. SA-TSDE	37	2.14E-02

Table 15. The Wilcoxon signed rank test results for mean value comparison.

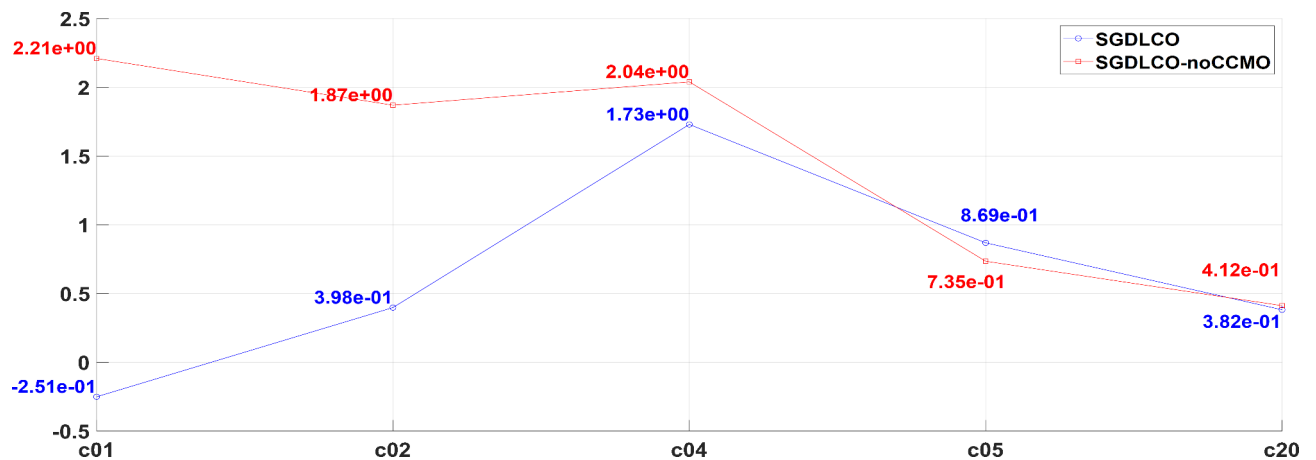


Fig. 7. Comparison results (mean values) among SGDLCO and SGDLCO-noCCMO on five 10-D test problems from IEEE CEC 2017.

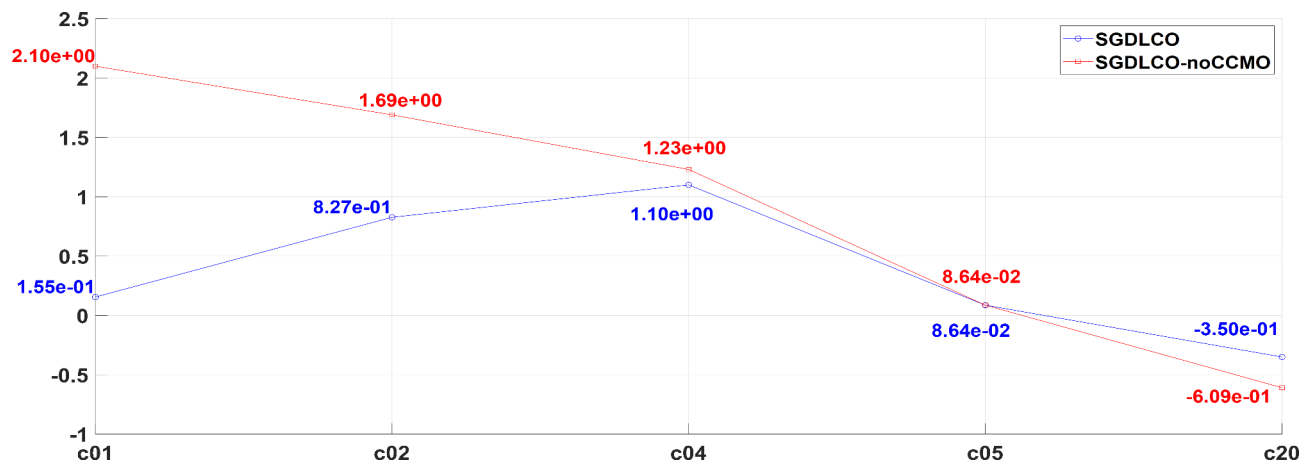


Fig. 8. Comparison results (std values) among SGDLCO and SGDLCO-noCCMO on five 10-D test problems from IEEE CEC 2017.

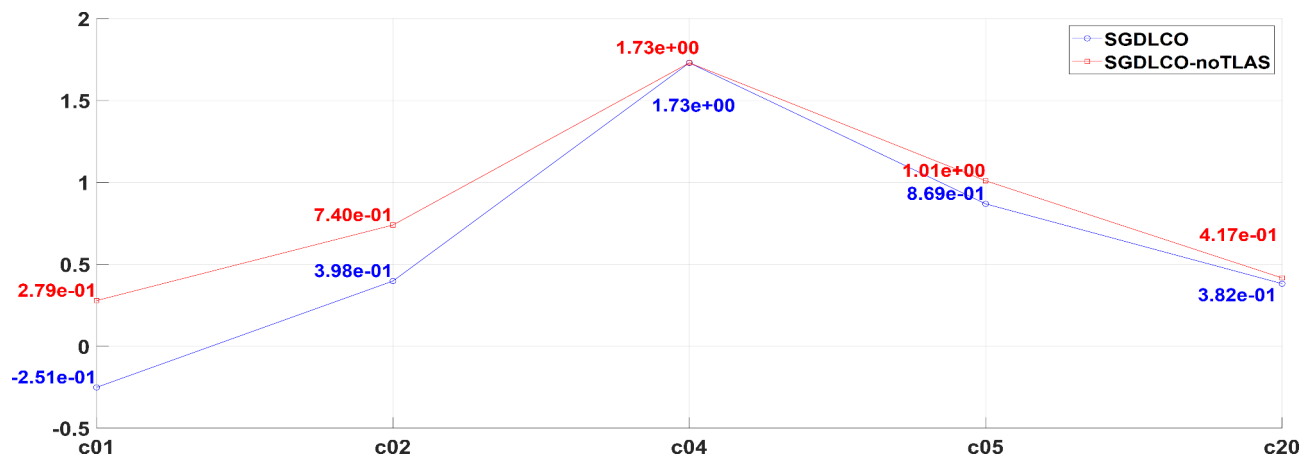


Fig. 9. Comparison results (mean values) among SGDLCO and SGDLCO-noTLAS on five 10-D test problems from IEEE CEC 2017.

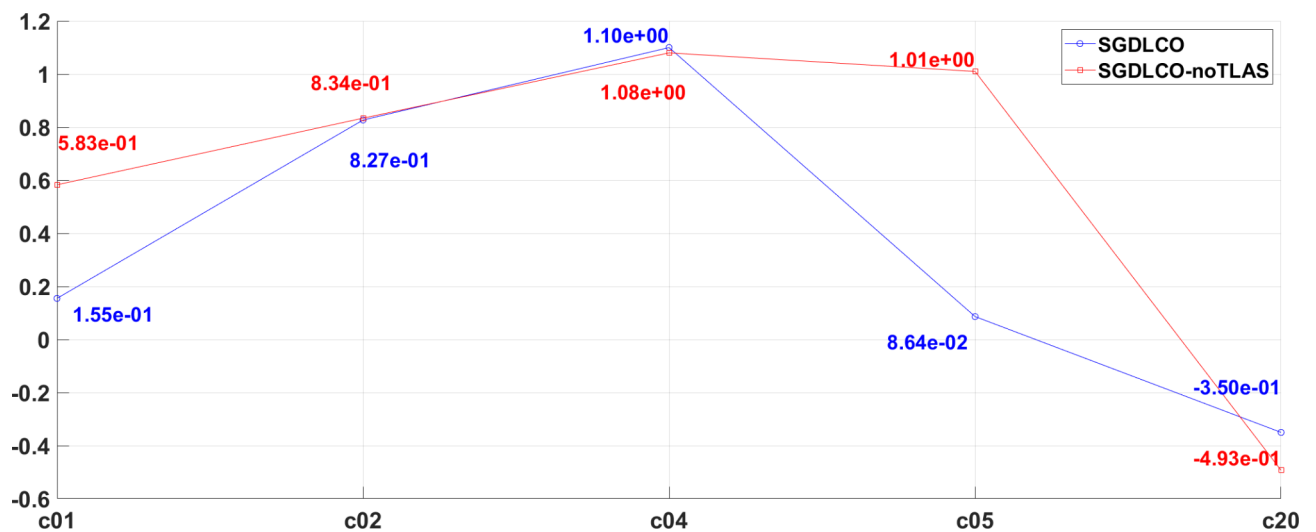


Fig. 10. Comparison results (std values) among SGDLCO and SGDLCO-noTLAS on five 10-D test problems from IEEE CEC 2017.

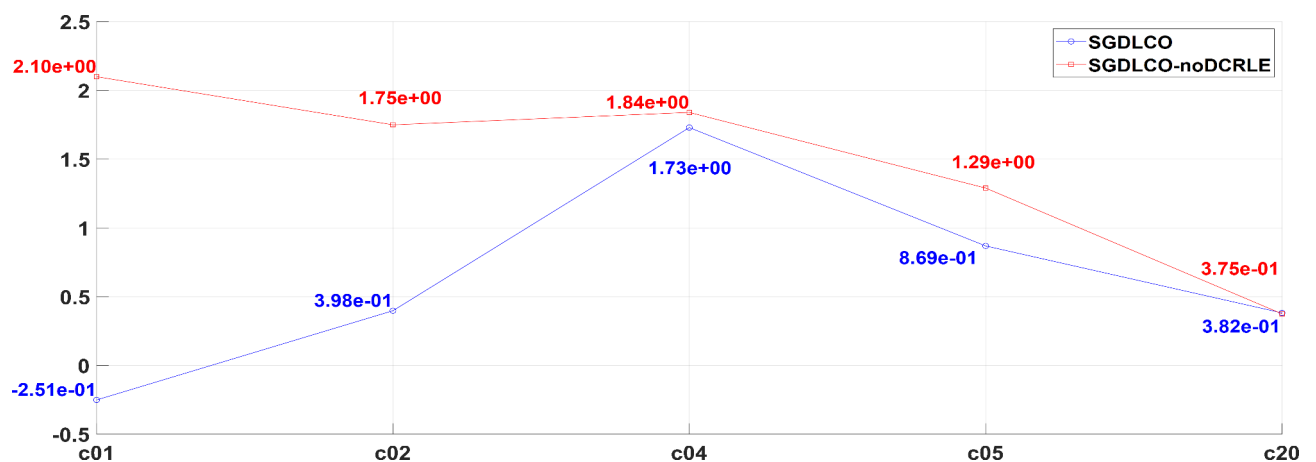


Fig. 11. Comparison results (mean values) among SGDLCO and SGDLCO-noDCRLE on five 10-D test problems from IEEE CEC 2017.

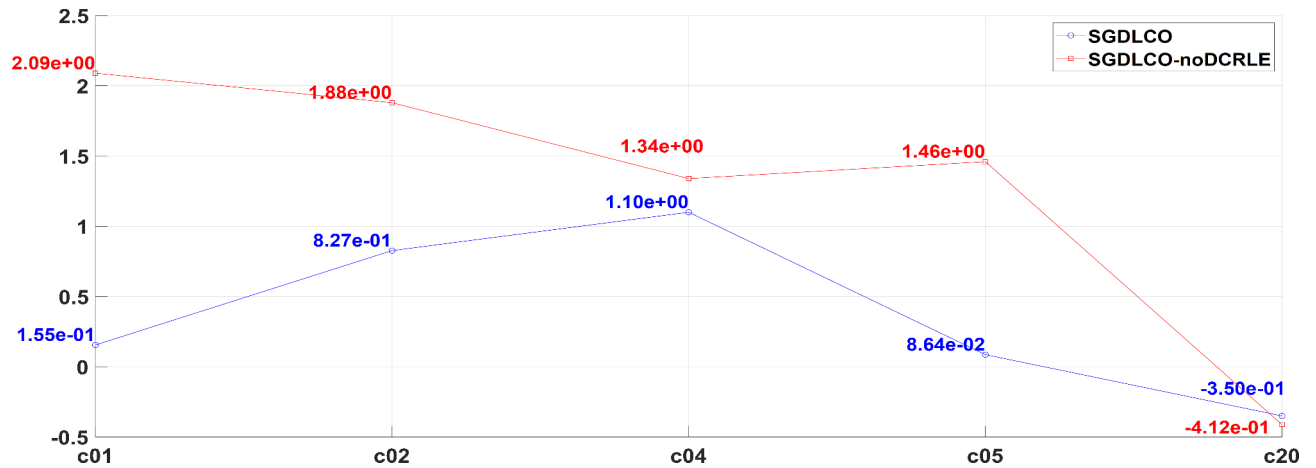


Fig. 12. Comparison results (std values) among SGDLCO and SGDLCO-noDCRLE on five 10-D test problems from IEEE CEC 2017.

searches, concentrating the search range on a smaller, more promising area, thereby making more efficient use of limited computational resources.

Conclusion

This paper introduces a surrogate-assisted global and distributed local collaborative optimization algorithm for expensive constrained optimization problems, named SGDLCO. The algorithm operates in two phases during each iteration: global surrogate-assisted collaborative evolution and distributed local surrogate-assisted search. In the global surrogate-assisted collaborative evolution phase, to fully utilize the positive guidance information from better solutions and the negative guidance information from poorer solutions, and to alleviate the pre-screening pressure on the surrogate model, the global candidate set is generated through classification cooperative mutation operation. In the distributed local surrogate-assisted phase, to effectively allocate uniformly distributed local searches, this paper designs a distributed central region local exploration method that uses affinity propagation clustering and mathematical modeling to identify uniformly distributed central solutions. For different clustering situations, targeted mathematical modeling methods are used to reselect central points for well-distributed local searches. Additionally, distance constraints are considered during the local search process to maintain search diversity and prevent the algorithm from getting trapped in local optima. To alleviate the selection pressure on the population, this paper designs a three-layer adaptive selection strategy. Two diversity-based fitness functions are formulated to screen candidate solutions from global or local candidate sets. This approach ensures the quality of the population while generating a more exploratory population, thereby significantly increasing the probability of TLAS finding the global optimum in the solution space and avoiding local optima. Compared to other algorithms, SGDLCO demonstrates significant advantages in handling expensive constraint optimization problems. It reduces the blind exploration in the global search stage and effectively avoids excessive redundancy in the local search. This significantly increases the likelihood of finding the global optimum and prevents the algorithm from getting trapped in local optima.

Furthermore, the design philosophy and methodology of SGDLCO provide a solid foundation for future algorithm extensions and improvements, particularly in addressing multi-objective and high-dimensional problems, where it is expected to achieve superior performance. Therefore, how to effectively extend the current proposed SGDLCO for solving these problems is an important future work.

Data availability

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Received: 30 November 2024; Accepted: 1 January 2025

Published online: 11 January 2025

References

1. Storn, R. & Price, K. Differential Evolution - A simple and efficient heuristic for Global Optimization over continuous spaces. *J. Glob Optim.* **11** <https://doi.org/10.1023/A:1008202821328> (1997).
2. Rao, R. V., Savsani, V. J. & Vakharia, D. P. Teaching-learning-based optimization: a novel method for constrained Mechanical Design optimization problems. *CAD Comput. Aided Des.* **43** <https://doi.org/10.1016/j.cad.2010.12.015> (2011).
3. Kennedy, J. & Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks - Conference Proceedings; Vol. 4. (1995).
4. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **69** <https://doi.org/10.1016/j.advengsoft.2013.12.007> (2014).
5. Wang, Y., Yin, D. Q., Yang, S. & Sun, G. Global and local surrogate-assisted Differential Evolution for Expensive Constrained optimization problems with inequality constraints. *IEEE Trans. Cybern.* **49** <https://doi.org/10.1109/TCYB.2018.2809430> (2019).

6. Dang, Q., Zhang, G., Wang, L., Yang, S. & Zhan, T. Hybrid IoT device selection with knowledge transfer for Federated Learning. *IEEE Internet Things J.* **11**, 12216–12227. <https://doi.org/10.1109/IIOT.2023.3334018> (2024).
7. Li, Y., Feng, X. & Yu, H. Solving high-dimensional expensive Multiobjective optimization problems by adaptive decision Variable Grouping. *IEEE Trans. Evol. Comput.* **PP**, 1 <https://doi.org/10.1109/TEVC.2024.3383095> (2024).
8. Ming, M., Wang, R., Zha, Y. & Zhang, T. Multi-objective optimization of hybrid renewable energy system using an enhanced multi-objective evolutionary algorithm. *Energies* **10** <https://doi.org/10.3390/en10050674> (2017).
9. Zhu, X. et al. Agent-based dynamic scheduling for Earth-Observing tasks on multiple airships in Emergency. *IEEE Syst. J.* **10** <https://doi.org/10.1109/JSYST.2014.2327069> (2016).
10. Şevkli, A. Z., Güler, B. A. & Multi-Phase Oscillated variable Neighbourhood Search Algorithm for a real-World Open Vehicle Routing Problem. *Appl. Soft Comput. J.* **58** <https://doi.org/10.1016/j.asoc.2017.04.045> (2017).
11. Dang, Q., Zhang, G., Wang, L., Yang, S. & Zhan, T. A. Generative adversarial networks Model Based Evolutionary Algorithm for Multimodal Multi-objective optimization. *IEEE Trans. Emerg. Top. Comput. Intell.* 1–10. <https://doi.org/10.1109/TETCI.2024.3397996> (2024).
12. Branke, J., Asafuddoula, M., Bhattacharjee, K. S. & Ray, T. Efficient use of partially converged simulations in Evolutionary optimization. *IEEE Trans. Evol. Comput.* **21** <https://doi.org/10.1109/TEVC.2016.2569018> (2017).
13. Grimalaccia, F., Mussetta, M. & Zich, R. E. Genetical swarm optimization: self-adaptive hybrid evolutionary algorithm for Electromagnetics. *IEEE Trans. Antennas Propag.* **55** <https://doi.org/10.1109/TAP.2007.891561> (2007).
14. Jiao, R., Sun, Y., Sun, J., Jiang, Y. & Zeng, S. Antenna Design using dynamic Multiobjective Evolutionary Algorithm. *IET Microwaves Antennas Propag.* **12** <https://doi.org/10.1049/iet-map.2018.5298> (2018).
15. Zhou, J. et al. An Infeasible Solutions Diversity Maintenance Epsilon Constraint Handling Method for Evolutionary Constrained Multiobjective optimization. *Soft Comput.* **25** <https://doi.org/10.1007/s00500-021-05880-5> (2021).
16. Zhang, Q., Li, H. & MOEA/D: A Multiobjective Evolutionary Algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11** <https://doi.org/10.1109/TEVC.2007.892759> (2007).
17. Wang, B. C., Li, H. X., Feng, Y. & Shen, W. J. An adaptive fuzzy penalty method for constrained evolutionary optimization. *Inf. Sci. (Ny)*. **571** <https://doi.org/10.1016/j.ins.2021.03.055> (2021).
18. Ang, K. M., Lim, W. H., Isa, N. A. M., Tiang, S. S. & Wong, C. H. A constrained Multi-swarm particle Swarm optimization without velocity for constrained optimization problems. *Expert Syst. Appl.* **140** <https://doi.org/10.1016/j.eswa.2019.112882> (2020).
19. Rahimi, I., Gandomi, A. H., Chen, F. & Mezura-Montes, E. A. Review on Constraint Handling Techniques for Population-Based Algorithms: From Single-Objective to Multi-Objective Optimization. *Arch. Comput. Methods Eng.* **30**, (2023).
20. Rosenbrock, H. H. An automatic method for finding the Greatest or least value of a function. *Comput. J.* **3** <https://doi.org/10.1093/comjnl/3.3.175> (1960).
21. Galletly, J. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. *Kybernetes* **27**, (1998).
22. Jan, M. A. & Khanum, R. A. A study of two penalty-parameterless constraint handling techniques in the Framework of MOEA/D. *Appl. Soft Comput. J.* **13** <https://doi.org/10.1016/j.asoc.2012.07.027> (2013).
23. Mallipeddi, R. & Suganthan, P. N. Ensemble of constraint handling techniques. *IEEE Trans. Evol. Comput.* **14** <https://doi.org/10.1109/TEVC.2009.2033582> (2010).
24. Liu, B., Zhang, Q. & Gielen, G. G. E. A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans. Evol. Comput.* **18** <https://doi.org/10.1109/TEVC.2013.2248012> (2014).
25. Glotic, A., Glotic, A., Kitak, P., Pihler, J. & Tícar, I. Parallel self-adaptive Differential Evolution Algorithm for solving short-term Hydro Scheduling Problem. *IEEE Trans. Power Syst.* **29** <https://doi.org/10.1109/TPWRS.2014.2302033> (2014).
26. Liu, Y., Yang, Z., Xu, D., Qiu, H. & Gao, L. A. Kriging-assisted double Population Differential evolution for mixed-integer Expensive Constrained optimization problems with mixed constraints. *Swarm Evol. Comput.* **84** <https://doi.org/10.1016/j.swevo.2023.101428> (2024).
27. Li, X. et al. Limit state kriging modeling for reliability-based design optimization through classification uncertainty quantification. *Reliab. Eng. Syst. Saf.* **224** <https://doi.org/10.1016/j.res.2022.108539> (2022).
28. Dong, H., Wang, P., Fu, C. & Song, B. Kriging-assisted teaching-learning-based optimization (KTLBO) to solve computationally expensive constrained problems. *Inf. Sci. (Ny)*. **556** <https://doi.org/10.1016/j.ins.2020.09.073> (2021).
29. Yang, Z. et al. Incorporating gradient information into dimension perturbation mutation for high-dimensional expensive optimization. *Swarm Evol. Comput.* **84** <https://doi.org/10.1016/j.swevo.2023.101446> (2024).
30. Yang, Z. et al. MOEA/D for Expensive constrained Multi-objective optimization. *Inf. Sci. (Ny)*. **639** <https://doi.org/10.1016/j.ins.2023.119016> (2023).
31. Chu, S. et al. Explicit topology optimization of Novel Polyline-based core sandwich structures using surrogate-assisted evolutionary algorithm. *Comput. Methods Appl. Mech. Eng.* **369** <https://doi.org/10.1016/j.cma.2020.113215> (2020).
32. Liu, Y., Yang, Z., Xu, D., Qiu, H. & Gao, L. A. Surrogate-assisted Differential Evolution for Expensive constrained optimization problems involving mixed-integer variables. *Inf. Sci. (Ny)*. **622** <https://doi.org/10.1016/j.ins.2022.11.167> (2023).
33. Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* **20** <https://doi.org/10.1023/A:1022627411411> (1995).
34. Li, C., Fang, H. & Gong, C. Expensive Inequality Constraints Handling Methods Suitable for Dynamic Surrogate-Based Optimization. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings; (2019).
35. Li, G. & Zhang, Q. Multiple penalties and multiple local surrogates for Expensive Constrained optimization. *IEEE Trans. Evol. Comput.* **25** <https://doi.org/10.1109/TEVC.2021.3066606> (2021).
36. Lu, C. & Paulson, J. A. No-Regret Bayesian Optimization with Unknown Equality and Inequality Constraints Using Exact Penalty Functions. In Proceedings of the IFAC-PapersOnLine; ; Vol. 55. (2022).
37. Miranda-Varela, M. E. & Mezura-Montes, E. Constraint-handling techniques in surrogate-assisted evolutionary optimization. An empirical study. *Appl. Soft Comput. J.* **73** <https://doi.org/10.1016/j.asoc.2018.08.016> (2018).
38. Gu, Q., Wang, Q., Li, X., Li, X. A. & Surrogate-Assisted Multi-objective particle Swarm optimization of Expensive constrained Combinatorial optimization Problems[Formula Presented]. *Knowledge-Based Syst.* **223** <https://doi.org/10.1016/j.knsys.2021.107049> (2021).
39. Huang, P., Wang, H. & Ma, W. Stochastic Ranking for Offline Data-Driven Evolutionary Optimization Using Radial Basis Function Networks with Multiple Kernels. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence, SSCI ; 2019. (2019).
40. Yang, Z. et al. Surrogate-assisted classification-collaboration Differential Evolution for Expensive Constrained optimization problems. *Inf. Sci. (Ny)*. **508** <https://doi.org/10.1016/j.ins.2019.08.054> (2020).
41. Zhang, Z., Wang, Y., Liu, J., Sun, G. & Tang, K. A. Two-phase kriging-assisted evolutionary algorithm for Expensive Constrained Multiobjective optimization problems. *IEEE Trans. Syst. Man. Cybern. Syst.* **54**, 4579–4591. <https://doi.org/10.1109/TSMC.2024.3383219> (2024).
42. Gu, Q., Wang, X., Wang, D., Liu, D. A. & Kriging-Assisted Multi-stage Evolutionary Algorithm for Expensive many-objective optimization problems. *Struct. Multidiscip. Optim.* **67** <https://doi.org/10.1007/s00158-024-03748-4> (2024).
43. Song, Z., Wang, H., He, C., Jin, Y. A. & Kriging-Assisted Two-archive evolutionary algorithm for expensive many-objective optimization. *IEEE Trans. Evol. Comput.* **25** <https://doi.org/10.1109/TEVC.2021.3073648> (2021).

44. Wei, F. F. et al. RBF-Assisted Differential Evolution for Distributed Expensive Constrained Optimization. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); ; Vol. 13824 LNAI. (2023).
45. Bai, F., Zou, D., Wei, Y. A. & Surrogate-Assisted Evolutionary algorithm with clustering-based sampling for high-dimensional expensive Blackbox optimization. *J. Glob Optim.* **89** <https://doi.org/10.1007/s10898-023-01343-3> (2024).
46. Chen, G. et al. Radial basis function surrogate model assisted evolutionary algorithm for high-dimensional expensive optimization problems. *Appl. Soft Comput.* **116** <https://doi.org/10.1016/j.asoc.2021.108353> (2022).
47. Horaguchi, Y. & Nakata, M. High-Dimensional Expensive Optimization by Classification-Based Multiobjective Evolutionary Algorithm with Dimensionality Reduction. In Proceedings of the 2023 62nd Annual Conference of the Society of Instrument and Control Engineers, SICE ; 2023. (2023).
48. Shi, R., Liu, L., Long, T. & Liu, J. Sequential radial basis function using support Vector Machine for Expensive Design optimization. *AIAA J.* **55** <https://doi.org/10.2514/1.J054832> (2017).
49. Liao, Y. S., Tan, K. C., Hu, J., Qiu, X. & Gee, S. B. Machine Learning Enhanced Multi-Objective Evolutionary Algorithm Based on Decomposition. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); ; Vol. 8206 LNCS. (2013).
50. Lin, L. et al. Classification model-based assisted Preselection and Environment Selection Approach for Evolutionary Expensive Bilevel optimization. *Appl. Intell.* **53** <https://doi.org/10.1007/s10489-023-04916-7> (2023).
51. Regis, R. G. Particle swarm with radial basis function surrogates for expensive black-box optimization. *J. Comput. Sci.* **5** <https://doi.org/10.1016/j.jocs.2013.07.004> (2014).
52. Yu, H., Tan, Y., Sun, C., Zeng, J. A. & Generation-Based Optimal restart strategy for surrogate-assisted Social Learning particle Swarm optimization. *Knowledge-Based Syst.* **163** <https://doi.org/10.1016/j.knsys.2018.08.010> (2019).
53. Zheng, N., Wang, H. & Yuan, B. An adaptive model switch-based surrogate-assisted evolutionary algorithm for noisy expensive multi-objective optimization. *Complex. Intell. Syst.* **8** <https://doi.org/10.1007/s40747-022-00717-6> (2022).
54. Yu, H., Kang, L., Tan, Y., Zeng, J. & Sun, C. A. Multi-model assisted Differential Evolution Algorithm for computationally expensive optimization problems. *Complex. Intell. Syst.* **7** <https://doi.org/10.1007/s40747-021-00421-x> (2021).
55. Li, E. An adaptive surrogate assisted Differential Evolutionary Algorithm for High Dimensional constrained problems. *Appl. Soft Comput.* **85** <https://doi.org/10.1016/j.asoc.2019.105752> (2019).
56. Tang, Z., Xu, L. & Luo, S. Adaptive dynamic surrogate-assisted evolutionary computation for high-fidelity optimization in Engineering. *Appl. Soft Comput.* **127** <https://doi.org/10.1016/j.asoc.2022.109333> (2022).
57. Couckuyt, I., De Turck, F., Dhaene, T. & Gorissen, D. Automatic Surrogate Model Type Selection during the Optimization of Expensive Black-Box Problems. In Proceedings of the Proceedings - Winter Simulation Conference; (2011).
58. Zhao, F. et al. Hybrid iterated local search algorithm with adaptive perturbation mechanism by Success-History based parameter adaptation for Differential Evolution (SHADE). *Eng. Optim.* **52** <https://doi.org/10.1080/0305215X.2019.1595611> (2020).
59. Dang, Q., Yang, S., Liu, Q. & Ruan, J. Adaptive and communication-efficient zeroth-order optimization for distributed internet of things. *IEEE Internet Things J.* **11**, 37200–37213. <https://doi.org/10.1109/JIOT.2024.3441691> (2024).
60. Frey, B. J. & Dueck, D. Clustering by passing messages between data points. *Sci. (80-)*. **315** <https://doi.org/10.1126/science.1136800> (2007).
61. Liang, J. J. et al. Problem definitions and evaluation criteria for the CEC 2006 Special Session on Constrained Real-Parameter optimization. *Tech. Rep.* (2005).
62. Mallipeddi, R. & Suganthan, P. *Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization*; ; (2010).
63. Awad, N. H., Ali, M. Z., Liang, J., Qu, B. Y. & Suganthan, P. N. Problem definitions and evaluation criteria for the CEC 2017 Special Session and Competition on Real-Parameter optimization. *Nanyang Technol. Univ. Singap. Tech. Rep.* (2016).
64. Wu, G. et al. Ensemble of Differential Evolution variants. *Inf. Sci. (Ny)*. **423** <https://doi.org/10.1016/j.ins.2017.09.053> (2018).
65. Powell, M. J. D. Restart procedures for the Conjugate Gradient Method. *Math. Program.* **12** <https://doi.org/10.1007/BF01593790> (1977).
66. Wang, Y., Cai, Z. & Zhang, Q. Differential Evolution with Composite Trial Vector Generation Strategies and Control parameters. *IEEE Trans. Evol. Comput.* **15** <https://doi.org/10.1109/TEVC.2010.2087271> (2011).
67. Wang, B. C., Li, H. X., Li, J. P. & Wang, Y. Composite Differential Evolution for constrained evolutionary optimization. *IEEE Trans. Syst. Man. Cybern. Syst.* **49** <https://doi.org/10.1109/TSMC.2018.2807785> (2019).
68. Wang, Y., Li, J. P., Xue, X. & Wang, B. C. Utilizing the correlation between constraints and objective function for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.* **24** <https://doi.org/10.1109/TEVC.2019.2904900> (2020).
69. Wang, Y., Wang, B. C., Li, H. X. & Yen, G. G. Incorporating objective function information into the feasibility rule for constrained evolutionary optimization. *IEEE Trans. Cybern.* **46** <https://doi.org/10.1109/TCYB.2015.2493239> (2016).
70. Wang, B. C., Li, H. X., Zhang, Q. & Wang, Y. Decomposition-based Multiobjective optimization for constrained evolutionary optimization. *IEEE Trans. Syst. Man. Cybern. Syst.* **51** <https://doi.org/10.1109/TSMC.2018.2876335> (2021).
71. Rahi, K. H., Singh, H. K. & Ray, T. Partial evaluation strategies for expensive evolutionary constrained optimization. *IEEE Trans. Evol. Comput.* **25** <https://doi.org/10.1109/TEVC.2021.3078486> (2021).
72. Liu, Y. et al. Two-stage Differential Evolution for Expensive Constrained optimization. *IEEE Trans. Emerg. Top. Comput. Intell.* **7** <https://doi.org/10.1109/TETCI.2023.3240221> (2023).
73. Derrac, J., García, S., Molina, D. & Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing Evolutionary and Swarm Intelligence algorithms. *Swarm Evol. Comput.* **1** <https://doi.org/10.1016/j.swevo.2011.02.002> (2011).

Acknowledgements

We would like to acknowledge Prof. Kamrul Hasan Rahi, Prof. Hemant Kumar Singh and Prof. Tapabrata Ray for sharing the original codes of SPAREA in the website: <https://codeocean.com/capsule/5554642/tree/v1>. Also, we would like to acknowledge both Prof. Yong Wang, Prof. Yuanhao Liu, and Prof. Genghui Li for sharing the sources codes of GLoSADE, SA-TSDE, and MPMLS.

This research is supported by the Topology optimization design of multi-scale composite porous metamaterials [grant numbers BSKYCZX 2023-07]; Young Talent Cultivation Innovation Fund Project of Nanchang University [grant numbers 9302-03740040]; Development of Multi-Source Micro-grid: Intelligent Control, Efficient Thermal Management, Noise Reduction, and Infrared Stealth Technology [grant numbers 20223AAE02012]; Key Technology Research on High-Power Hydrogen Fuel Cell Metal Ultra-Thin Bipolar Plates for Multi-Source Energy Equipment [grant numbers 20232BCJ22058]; Nanchang County (Xiaolan Economic Development Zone) for the support of the 2023 “Unveiling the List and Taking the Lead” project (Multi-channel Low-noise Gas Turbine Generator Set).

Author contributions

Conceptualization, X.L. and Z.Y.; methodology, X.L., Z.Y. and S.H.; software, X.L., Z.Y. and S.H.; validation, X.L., Z.Y. and S.H.; formal analysis, X.L. and Z.Y.; investigation, X.L. and J.L.; resources, J.L. and Z.Y.; data curation, X.L. and Z.Y.; writing—original draft preparation, X.L., Z.Y. and S.H.; writing—review and editing, X.L., Z.Y. and S.H.; visualization, J.Y.; supervision, J.X., J. H., X.F., and Z.Y.; project administration, J.L. and Z.Y.; funding acquisition, J.L., X.F., and Z.Y. All authors have read and agreed to the published version of the manuscript.

Declarations

Conflict of 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.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-85233-6>.

Correspondence and requests for materials should be addressed to Z.Y. or S.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025