A Surrogate-Assisted Constrained Optimization Evolutionary Algorithm by Searching Multiple Kinds of Global and Local Regions

Yong Zeng, Yuansheng Cheng, and Jun Liu

***Abstract*—This article proposes a surrogate-assisted evolutionary algorithm to tackle expensive inequality-constrained optimization problems through global exploration and local exploitation. The algorithm begins with an exploration stage that involves sampling in three kinds of global regions: 1) the feasible region; 2) the better-objective region; and 3) the converging region. Specifically, sampling in the uncertain feasible region mitigates issues caused by inaccurate objective surrogates. In addition, sampling in the uncertain region containing better-objective values than the current best-feasible solution reduces the risk of missing the global optimum due to inaccurate constraint surrogates. Moreover, sampling in the converging region facilitates quick convergence to the global feasible optimum. Following the exploration stage, promising feasible and infeasible solutions are further refined using local surrogate-based search strategies. To address the risk of missing the global optimum resulting from limited local region scope, the regions are adaptively extended if predicted infill points lie on the boundary. If an infill point is determined to showcase a better-objective value after accurate evaluation, a rewarding local search is performed within the local region. This exploration-exploitation process iterates until the computation budget is exhausted. Experimental results demonstrate that the proposed algorithm outperforms the selected state-of-the-art algorithms on the majority of tested problems.**

***Index Terms*—Differential evolution (DE), expensive constrained optimization, global and local search, surrogate assisted evolutionary algorithm.**

I. Introduction

An OPTIMIZATION problem with inequality constraints can be formulated as follows:

$$\begin{aligned} &\min\ f(\mathbf{x}) \\ &\text{s.t.}\ \ g_i(\mathbf{x}) \le 0, i = 1, 2, \dots, p \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the decision vector containing D variables, $f(\mathbf{x})$ is the objective function to be minimized, and $g_i(\mathbf{x})$ is the ith inequality constraint, p is the number of inequality constraints. When the objective function and the constraints are expensive to be evaluated (e.g., calculation-intensive), the problem is called an expensive constrained optimization problem (ECOP). In this work, it is assumed that a single evaluation can provide both the objective and constraint values simultaneously. Typically, a restricted number of function evaluations are available when dealing with ECOPs. Therefore, devising algorithms that can robustly and efficiently obtain the global feasible optimum becomes a pivotal concern.

Surrogate-assisted evolutionary algorithms (SAEAs) have proven to be highly effective in addressing such problems [1], [2]. In SAEAs, surrogate techniques [3], [4] are adopted to assist the evaluation of the solutions. To mitigate the demand for expensive evaluations, three popular approaches for integrating surrogates into evolutionary algorithms can be found in the research: 1) evolution control; 2) surrogate-assisted prescreening, and 3) surrogated-based local search. In the evolution control approach [5], an EA is executed based on surrogates. Throughout the evolution process, individual-based or generation-based control methods are adopted to select some solutions for expensive evaluations. In the line of surrogate-assisted prescreening methods [6], [7], a number of candidate offspring are generated first. Subsequently, these offspring are pre-evaluated using surrogates, and based on specific selection rules, promising solutions are chosen for further expensive evaluations. Regarding surrogate-based local search methods [8], surrogates are first constructed within some specific local regions. An iterative search is then performed based on these surrogates to identify the best solution within each local region. These solutions are subsequently re-evaluated using the original objective function and constraints. In these methods, three imperative aspects can be observed in SAEAs, i.e., the construction of surrogates, the generation of candidate solutions, and the selection of solutions for expensive evaluations.

Generally, the surrogates used in SAEAs can be constructed globally or locally. Global surrogates can capture the overall trend of the functions in the problem, making them suitable for exploring the global search space [9]. However, it is worth noting that most of the existing methods construct global surrogates using all the truly-evaluated solutions [10], [11], [12], which might be time-consuming. Moreover, if the function values within the training set exhibit extreme differences, the prediction accuracy of the surrogate may decrease [13], [14]. In contrast, although utilizing only a portion of the available data may result in some loss of

Manuscript received 6 June 2023; revised 13 August 2023 and 6 November 2023; accepted 18 December 2023. Date of publication 25 December 2023; date of current version 31 January 2025. This article was approved by Associate Editor S. M. Elsayed. *(Corresponding author: Jun Liu.)*

The authors are with the School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yongzeng@hust.edu.cn; yscheng@hust.edu.cn; hustlj@hust.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TEVC.2023.3346435, provided by the authors.

Digital Object Identifier 10.1109/TEVC.2023.3346435

information about the original function, it can save costs, especially when the complexity of modeling is high [15]. In addition, training the global surrogates using a subset of the solutions may improve the prediction accuracy [16]. For instance, Song et al. [17] proposed a framework involving data selection, search intensity adjustment, and infill sampling criterion (DSI) for ECOPs. By selectively choosing different solutions for modeling the objective and constraints, the accuracy of the surrogates can be improved [17]. However, the selection strategies employed in DSI primarily focus on surrogate accuracy within the vicinity of the current population. Consequently, if the population loses diversity, it becomes challenging to escape from local search regions. Compared to global surrogates, it has been observed that local surrogates tend to achieve higher accuracy in approximating functions within local regions [10]. Consequently, exploitation strategies often incorporate local search procedures that leverage local surrogates [18], [19]. Besides their improved predictive performance in local regions, local surrogates also exhibit the advantage of requiring a comparatively smaller number of data points for modeling. This serves to alleviate the computational burden associated with the modeling process [20], [21]. Nevertheless, it is important to note that solely relying on local surrogates without the implementation of additional diversity maintenance strategies may result in the population becoming trapped within local regions. For instance, Eriksson et al. [22] proposed to construct local Gaussian process models (also known as Kriging models) and maintain multiple trust regions simultaneously. This approach was subsequently extended to address ECOPs [23]. The experimental results demonstrate highly promising performance of this method. However, this method favors feasible solutions over infeasible ones. When multiple feasible regions exist and the global optimum is in a small feasible region, the search may stagnate in a local feasible region.

In addition to surrogate modeling, the method of obtaining candidate offspring for expensive evaluation also plays a crucial role in algorithm performance. Generally, there are two approaches to obtaining candidate offspring: 1) one-shot manner and 2) iterative manner. In the one-shot manner, a specific number of offspring are generated first and evaluated using surrogates. Subsequently, a few of the best-performing offspring are selected and reevaluated using the original objective function and constraints (individual-based evolution control and surrogate-assisted prescreening) [20], [24]. Two key aspects are involved in the offspring generation step: 1) selecting which solutions as parents for generating offspring and 2) determining the genetic operators (such as mutation and crossover) to be applied to these parents. Typically, the parents are selected from the current population, and genetic operators like differential evolution (DE) [25], polynomial mutation, and simulated binary crossover [26] are applied to generate candidate offspring. For example, Zan et al. [11] introduced a classification-collaboration approach (SACCDE) to partition the individuals within the population into two distinct groups. Subsequently, two offspring generation strategies in DE, DE/best/2/bin and DE/rand/2/bin, are modified according to the classification results to generate many candidate offspring. By incorporating information from the best solution, this methodology achieved notable gains in convergence speed. However, the lack of consideration for model uncertainty in SACCDE renders it vulnerable to being trapped within local search regions, thereby exacerbating premature convergence issues. To mitigate this issue, several studies have focused on enhancing the exploration capability of algorithms by incorporating diversity-based offspring generation operators [10], [20]. For instance, Wang et al. [10] proposed a global and local surrogate-assisted differential evolution method (GloSADE) for ECOPs. In the global exploration stage of GloSADE, DE/current-to-random/1 is adopted to generate candidate offspring. However, it is important to note that the parents used in this strategy are solely derived from the current population and the distribution of the candidate offspring is heavily influenced by these parents. Consequently, this exploration strategy can be ineffective if the population experiences a loss of diversity.

In the iterative manner, candidate solutions are obtained by iteratively searching based on the surrogates. The obtained candidate solutions are then reevaluated expensively (generation-based evolution control and surrogate-based local search) [27], [28], [29]. For example, GloSADE adopts the interior point method to conduct searches on the local surrogates to accelerate the convergence. While this strategy effectively enhances efficiency, the best-performing solutions based on the feasibility rule (FR) [30] are determined as the local search locations. Consequently, this approach may result in ignoring promising yet infeasible solutions, which have proven to be valuable for addressing constrained problems [31].

Furthermore, the criteria utilized to determine the solutions for expensive evaluation prove to be crucial when ECOPs are encountered [17], [32]. Three kinds of criteria are widely used in the literature: 1) the best solutions according to the predictions; 2) the most uncertain solutions; and 3) solutions balancing the uncertainty and the predicted fitness. In the first category, the best solutions are selected solely based on the surrogate predictions. For instance, Regis [33] proposed to select the best solution using FR among a large number of candidate offspring. Although this method can improve search efficiency, the search can be led to the wrong regions due to inaccurate surrogates [34]. To alleviate this issue, Li and Zhang [20] proposed a multiple penalties and multiple local surrogates (MPMLSs) method. Multiple subregions are determined around each individual and the local surrogates are constructed in the local regions. Then, different penalty coefficients are utilized in each local region to facilitate the selection of solutions for expensive evaluation. By separating the search space into multiple subregions and employing varying penalty coefficients, population diversity is preserved, thereby mitigating the problem of becoming trapped in local regions. However, the misguiding issue by inaccurate surrogates cannot be avoided if the population has been led to a relatively small region. To improve the surrogate accuracy, the solutions with maximum uncertainties are reevaluated expensively in some studies [10], [17]. For example, the most uncertain solutions are selected for expensive evaluation in the global exploration stage in GloSADE. Although this method can provide a certain degree of exploration, the accuracy of the surrogates is improved in a relatively small region because the candidate offspring are generated from the current population. In the last category, criteria balancing exploration and exploitation are employed to determine the solutions for expensive evaluations [35], [36]. For instance, probability of feasibility (PoF) and expected improvement (EI) are combined to determine the expensively-evaluated points sequentially [37], [38]. However, these criteria also suffer from the issue that diverse candidate offspring cannot be effectively generated when the population loses diversity [39].

Based on the above brief review of methods for ECOPs, it can be observed that most of the existing works construct global surrogates using all history solutions. Very few algorithms have been developed to select a subset of solutions of interest for training global surrogates. In addition, the parents for generating candidate offspring typically come from the current population, making it hard to escape local search regions once the population loses diversity. Moreover, the search process tends to overly prioritize feasibility, leading to insufficient exploitation of promising infeasible solutions. To alleviate these issues, a SAEA by searching multiple kinds of global and local regions (MGRLR) is proposed in this study.

In particular, MGRLR consists of the exploration stage and the exploitation stage in each iteration. During the exploration stage, the predicted feasible region, the predicted better-objective region, and the converging region are sequentially explored. To achieve this goal, the training solutions for global surrogates, the parents for generating candidate offspring, and the solutions to be expensively evaluated are selected variously according to the exploration types. In this way, the exploration can be carried out effectively and efficiently. In the exploitation stage, both promising feasible and promising infeasible solutions are identified as local search locations. In this way, the global feasible optimum can be approached from both the feasible and infeasible sides. The contributions of this work are summarized as follows.

1) A novel framework exploring three types of global regions and exploiting two types of local regions for solving ECOPs is proposed. Through exploring the predicted feasible region, the issues brought by the inaccurate objective surrogate in the feasible region can be mitigated. By exploring the predicted better-objective region, the issues arising from inaccurate constraint surrogates in the better objective can be alleviated. Moreover, exploring the converging region is beneficial to enhance the convergence in the entire search space. To further facilitate convergence, surrogate-based local searches are carried out around both the promising feasible and promising infeasible solutions. In this way, the global optimum can be approached from the feasible and infeasible sides simultaneously.
2) A strategy is devised to select the training solutions for global surrogates, the parents for generating candidate offspring, and the solutions to be expensively evaluated according to different exploration types. Because only a subset of solutions is used for surrogate modeling, the influence of the outliers regarding the function values can be mitigated. By selecting various types of parents, diverse candidate offspring can be generated, thereby increasing the chances of escaping local search regions.
3) To enhance the efficiency of local searches, the local regions are adaptively extended according to search information. In particular, the local region is extended dimension-wise if the attained solution lies on the local region boundary. In this manner, the local optima in the local regions can be approached efficiently, thereby enhancing the overall efficiency of MGRLR.
4) To exploit the infeasible solution information further, a reward search mechanism is embedded in the framework. If the solution obtained in the first run of local search is determined to have a better-objective value, a reward search is carried out in the corresponding local region. Since the reward search is conducted in the local region that contains better-objective values, better-feasible solutions can be attained with a high probability.

The remainder of this article is organized as follows. Section II presents some necessary background techniques used in the proposed MGRLR. The details of MGRLR are introduced in Section III. Following the experimental study in Section IV, some conclusions are given in Section V.

II. Background

Some necessary techniques used in this study are presented in this section, i.e., radial basis function (RBF) modeling, several DE variants, and constraint handling techniques.

A. Radial Basis Function

Various types of surrogates have been used in SAEAs, e.g., Gaussian process model [40], polynomial regression [41], and RBF [42], [43]. In this study, RBF is adopted due to its efficiency and effectiveness in approximation [10].

An RBF with a linear polynomial tail for approximating a target function can be formulated as in

$$\widehat{f}(\mathbf{x}) = \sum_{k=1}^{N} w_k \varphi(\|\mathbf{x} - \mathbf{x}_k\|) + \beta_0 + \sum_{j=1}^{D} \beta_j x_j \tag{2}$$

where $\mathbf{x} = (x_{1,}x_2, \ldots, x_{j,}, \ldots, x_D)$ is the decision vector and $\mathbf{x}_k$ is the kth solution in the training set, N is the number of solutions in the set, $\varphi(\cdot)$ is the basis function, $\|\bullet\|$ denotes the Euclidean norm of a vector, w_k is the kth coefficient of the basis function, β_j is the jth coefficient in the polynomial, and $\widehat{f}(\mathbf{x})$ is the surrogate for the target function $f(\mathbf{x})$.

In this work, the basis function with the form $\varphi(r) = r^3$ is adopted, which is frequently used in [21] and [44]. In addition, the coefficients can be determined by solving the linear system in

$$\begin{pmatrix} \mathbf{\Phi} & \mathbf{P} \\ \mathbf{P}^{\mathbf{T}} & \mathbf{0}_{(D+1)\ \times(D+1)} \end{pmatrix} \begin{pmatrix} \mathbf{W} \\ \mathbf{B} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{0}_{D+1} \end{pmatrix} \tag{3}$$

where $\mathbf{\Phi}$ is a $N \times N$ matrix consisting of components $\Phi_{i,j} = \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|)$, $i, j = 1, 2, \ldots, N$. $\mathbf{P}$is a $N \times (D + 1)$ matrix with rows $\mathbf{P}_i = (1, x_{i,1}, x_{i,2}, \ldots, x_{i,D})$, $i = 1, 2, \ldots, N$.

$\mathbf{0}_{(D+1)\times(D+1)}$ and $\mathbf{0}_{D+1}$ are, respectively, a $(D+1)\times(D+1)$ matrix and a $(D+1)\times 1$ vector with zeros. $\mathbf{W} = (w_1, w_2, \ldots, w_N)^T$. $\mathbf{B} = (\beta_0, \beta_1, \ldots, \beta_D)^T$. $\mathbf{F} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_N))^T$ is the vector with real function values. To model multiple functions simultaneously, the formulation of (3) in the right hand can be extended by simply incorporating additional columns of function values following the same structure.

B. Differential Evolution

Abundant offspring generation strategies can be found in the EA community [45], [46], [47]. In this study, we employ three variants of DE to generate candidate offspring due to its simplicity and powerful search ability. The three mutation operators are formulated from (4)–(6).

1) DE/rand/1.

$$\mathbf{v} = \mathbf{x}_{r_1} + F \times \left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) \tag{4}$$

2) DE/random-to-random/1.

$$\mathbf{v} = \mathbf{x}_{\text{rand}} + F \times \left(\mathbf{x}_{r_1} - \mathbf{x}_{\text{rand}}\right) + F \times \left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) \tag{5}$$

3) DE/best/1

$$\mathbf{v} = \mathbf{x}_{\text{best}} + F \times \left(\mathbf{x}_{r_1} - \mathbf{x}_{r_2}\right) \tag{6}$$

where $\mathbf{x}_{\text{rand}}$, $\mathbf{x}_{r_1}$, $\mathbf{x}_{r_2}$, $\mathbf{x}_{r_3}$ are four randomly selected solutions from the set of parents. $\mathbf{x}_{\text{best}}$ is the best solution under some specific kind of rule, e.g., FR. F is the scaling factor.$\mathbf{v}$ is the mutant vector. After mutation, the crossover operator is conducted on the mutant vector obtained by (4) and (6) The binomial crossover is presented as follows:

$$u_j = \begin{cases} v_j, & \text{if } \text{rand}_j \le \text{CR or } j = j_{\text{rand}}, j = 1, \ldots, D \\ x_{k,j}, & \text{otherwise} \end{cases} \tag{7}$$

where u_j is the jth element of an individual $\mathbf{u}$. rand$_j$ is a random number with uniform distribution in the interval [0,1]. CR is the crossover probability. j_{rand} is a random integer ranging from 1 to D. k denotes r_1 or best. v_j and $x_{k,j}$ denote the jth element of $\mathbf{v}$ and $\mathbf{x}_k$, respectively. DE/random-to-random/1 is a variant of DE/current-to-random/1. Because the crossover operation is generally not implemented on $\mathbf{v}$ obtained by DE/current-to-random/1 [48], [49], $\mathbf{u} = \mathbf{v}$is adopted for $\mathbf{v}$ obtained by (5)

By using DE/rand/1 and DE/random-to-random/1, diverse solutions can be generated and exploration is expected. In contrast, DE/best/1 utilizes the information of the best solution found so far. Therefore, exploitation can be achieved.

C. Constraint Handling

When constrained optimization problems are encountered, the balance between feasibility and optimality is imperative. On the one hand, the search may be attracted to a large local feasible region if the feasibility is given too much focus. On the other hand, if no sufficient effort is put into feasibility, the final solution may be infeasible. Therefore, proper constraint handling techniques (CHTs) play a critical role in solving ECOPs. Frequently used CHTs include FR [30], penalty methods [50], multiobjective methods [51], stochastic sorting [52], and epsilon-relaxation methods [31], to name a few. In this study, FR is adopted in several steps. The FR [30] is given as follows.

1) Between two feasible solutions, the solution with better-objective value is preferred;
2) Feasible solutions are better than infeasible solutions;
3) Between two infeasible solutions, the solution with a smaller constraint violation (CV) is better.

CV of a solution is defined as in

$$\text{CV}(\mathbf{x}) = \sum_{i=1}^{p} \max(g_i(\mathbf{x}), 0). \tag{8}$$

p and $g_i(\mathbf{x})$ have the same meanings as in (1)

III. Proposed Method

In this section, the proposed MGRLR is introduced. First, the framework of MGRLR is given. Then, the details of MGRLR are presented.

A. MGRLR

As shown in Algorithm 1, MGRLR begins with the initialization of N solutions with the Latin hypercube sampling method (LHS) [4]. Then, the solutions are expensively evaluated. An archive is adopted to save all the solutions, including the decision variables and the corresponding objective and constraint values. The objective value of the best-feasible solution found so far is recorded as f_{best}. If no feasible solution has been found, f_{best} is set to be $+\infty$. The number of function evaluations that have been consumed is recorded as *FEs*. Afterward, the optimization procedure continues to iterate until the computational budget is exhausted.

During each iteration, the exploration and exploitation are carried out alternatively. Specifically, the exploration stage is executed in Algorithm 1 Lines 6–9. When conducting exploration, the type of exploration is first determined. Subsequently, a global search (Algorithm 2) is performed to obtain a solution for expensive evaluation. Further details regarding this process are introduced in Section III. B. Following the exploration stage, the locations for local search are identified. The exploitation stage is then carried out in Algorithm 1 Lines 11–16, which is elaborated in detail in Section III. C. Within each iteration of the exploitation stage, an initial local search (Algorithm 3) is conducted. Furthermore, after the true evaluation of the obtained solution, if it is determined to possess a better-objective value compared to the best-feasible solution found so far, a rewarding local search (Algorithm 3) is additionally executed to improve this solution further.

The MGRLR framework bears resemblance to several hierarchical exploration-exploitation algorithms [10], [53]. However, unlike existing studies where exploration and exploitation primarily occur within the neighborhood of the current population, MGRLR distinguishes itself by conducting exploration and exploitation in multiple kinds of global and local regions. The global regions encompass the predicted feasible region, better-objective region, and converging region.

Algorithm 1: MGRLR

Input: number of initial solutions N, maximum number of expensive function evaluations *maxFEs*, maximum number of solutions used for global surrogates N_G, maximum number of solutions used for local surrogates N_L
Output: the best solution $\mathbf{x}_{\text{best}}$ in $\mathbf{A}$

Initialize N solutions by LHS and evaluate the solutions expensively;
Save the solutions into an archive $\mathbf{A}$;
$f_{\text{best}} \leftarrow$ The objective value of the best feasible solution;
`//` $f_{\text{best}} \leftarrow +\infty$`; if no feasible solution exists`
$FEs \leftarrow N$;
while $FEs < maxFEs$ **do**
  `// Global exploration`
  for $i \leftarrow$ **to** N **do**
    Type $\leftarrow$ Determine the exploration type;
    $\mathbf{O}_i \leftarrow$ Global search with **Algorithm 2**($\mathbf{A}$, *Type*, N_G, N, f_{best});
    Update $\mathbf{A}$, f_{best}, and FEs;
  `// Local exploitation`
  $\mathbf{L} \leftarrow$ Determine the locations for local search;
  for $i \leftarrow$ **to** N **do**
    Type $\leftarrow$ Determine the exploration type;
    $\mathbf{O}_i \leftarrow$ Local search with **Algorithm 3**($\mathbf{A}, \mathbf{L}_i, N_L, f_{\text{best}}$);
    Update $\mathbf{A}$, f_{best}, and FEs;
    `// Reward Local exploitation`
    if $f(\mathbf{O}_i) \leq f_{\text{best}}$ **then**
      $\mathbf{O}_i \leftarrow$ Reward search with **Algorithm 3**($\mathbf{A}, \mathbf{O}_i, N_L, f_{\text{best}}$);
      Update $\mathbf{A}$, f_{best}, and FEs;

Meanwhile, the local regions encompass regions with better-objective values (arranged based on constraint violations) and regions with equal or worse-objective values (arranged based on objective values). For convenience, the details are introduced in the subsequent subsections.

B. Global Exploration

When surrogates are used in the EA, there are two potential reasons that lead to the search getting trapped in a local region. First, it is difficult to escape from a local region when the population loses diversity. Second, the surrogates may be misguiding. Specifically, promising solutions may be filtered out by inaccurate surrogates even if they have been successfully generated.

To address the first issue, the solutions distributed diversely in the restricted region (the feasible region or the better-objective region) are selected as parents for generating candidate offspring. Afterward, the most uncertain candidate offspring in the restricted region is chosen for expensive evaluation (predicted to be feasible or predicted to possess a better-objective value). Therefore, the misguiding issue caused by inaccurate surrogates can be alleviated. It deserves to point out that the candidate offspring are generated using the current population in most existing studies [10], [17]. Consequently, if the population loses diversity, the exploration ability of these algorithms decreases.

Algorithm 2: Global Search

Input: $\mathbf{A}$, *Type*, N_G, N, f_{best}
Output: the expensively evaluated solution $\mathbf{O}_s$

$n_1 \leftarrow \lfloor N_G/2 \rfloor$; $n_2 \leftarrow (N_G - n_1)$; $N_1 \leftarrow \lfloor N/2 \rfloor$; $N_2 \leftarrow (N - N_1)$;
$\mathbf{A}_1 \leftarrow$ Solutions with objective values f better than f_{best}; $\mathbf{A}_2 \leftarrow \mathbf{A} \backslash \mathbf{A}_1$;
switch *Type* **do**
  case *Feasible exploration*
    $\mathbf{D_T} \leftarrow n_1$-sparsest feasible and n_2-sparsest infeasible solutions;
    $\mathbf{M} \leftarrow N$-sparsest feasible solutions;
    Construct global surrogates with $\mathbf{D_T}$;
    $\mathbf{O} \leftarrow$ Generate candidate offspring with $\mathbf{M}$;
    $\mathbf{O}_s \leftarrow$ 1-sparsest solution in feasible $\mathbf{O}$;
  case *Better-objective exploration*
    $\mathbf{D_T} \leftarrow n_1$-sparsest solutions in $\mathbf{A}_1$ and n_2-best-f solutions in $\mathbf{A}_2$;
    $\mathbf{M} \leftarrow N$-sparsest solutions in $\mathbf{A}_1$;
    Construct global surrogates with $\mathbf{D_T}$;
    $\mathbf{O} \leftarrow$ Generate candidate offspring with $\mathbf{M}$;
    $\mathbf{O}_s \leftarrow$ 1-sparsest solution in better-f $\mathbf{O}$;
  case *Constrained optimization*
    $\mathbf{D_T} \leftarrow n_1$-best-$CV$ solutions in $\mathbf{A}_1$ and n_2-best-f solutions in $\mathbf{A}_2$;
    $\mathbf{M} \leftarrow N_1$-best-$CV$ solutions in $\mathbf{A}_1$ and N_2-best-f solutions in $\mathbf{A}_2$;
    Construct global surrogates with $\mathbf{D_T}$;
    $\mathbf{O} \leftarrow$ Generate candidate offspring with $\mathbf{M}$;
    $\mathbf{O}_1 \leftarrow$ Solutions with objective values better than f_{best} in $\mathbf{O}$;
    if $|\mathbf{A}_1| = 0$ *or* $|\mathbf{O}_1| = 0$ **then**
      $\mathbf{O}_s \leftarrow$ 1-best solution in $\mathbf{O}$ regarding the objective;
    else
      $\mathbf{O}_s \leftarrow$ 1-best solution in $\mathbf{O}_1$ based on FR;
Evaluate $\mathbf{O}_s$ expensively;
Initialize N solutions by LHS and evaluate the solutions expensively;
Save the solutions into an archive $\mathbf{A}$;
$f_{\text{best}} \leftarrow$ the objective value of the best feasible solution;
`//` $f_{\text{best}} \leftarrow +\infty$`; if no feasible solution exists`
$FEs \leftarrow N$;
while $FEs < maxFEs$ **do**
  `// Global exploration`
  for $i \leftarrow$ **to** N **do**
    Type $\leftarrow$ Determine the exploration type;
    $\mathbf{O}_i \leftarrow$ Global search with **Algorithm 2**($\mathbf{A}$, *Type*, N_G, N, f_{best});
    Update $\mathbf{A}$, f_{best}, and FEs;
  `// Local exploitation`
  $\mathbf{L} \leftarrow$ Determine the locations for local search;
  for $i \leftarrow$ **to** N **do**
    Type $\leftarrow$ Determine the exploration type;
    $\mathbf{O}_i \leftarrow$ Local search with **Algorithm 3**($\mathbf{A}, \mathbf{L}_i, N_L, f_{\text{best}}$);
    Update $\mathbf{A}$, f_{best}, and FEs;
    `// Reward Local exploitation`
    if $f(\mathbf{O}_i) \leq f_{\text{best}}$ **then**
      $\mathbf{O}_i \leftarrow$ Reward search with **Algorithm 3**($\mathbf{A}, \mathbf{O}_i, N_L, f_{\text{best}}$);
      Update $\mathbf{A}$, f_{best}, and FEs;

Algorithm 3: Local Search

Input: $\mathbf{A}, \mathbf{x}, N_L, f_{\text{best}}$
Output: the expensively evaluated solution $\mathbf{O}_s$
$k \leftarrow 0$; $k_{max} \leftarrow 1$;
$\mathbf{D_T} \leftarrow N_L$ nearest solutions around $\mathbf{x}$; `// Solutions from A`
$\mathbf{x}_{\min}, \mathbf{x}_{max} \leftarrow$ Vectors consisting of boundary values of solutions in $\mathbf{D_T}$;
while $k \le k_{max}$ **do**
  Construct local surrogates with $\mathbf{D_T}$;
  $n_b \leftarrow$ Number of solutions in $\mathbf{D_T}$ with objective values better than f_{best};
  if $n_b > 0$ *or no feasible solution* **then**
    $\mathbf{O}_s \leftarrow$ Constrained local search;
  else
    $\mathbf{O}_s \leftarrow$ Unconstrained local search;
  if $\mathbf{O}_s$ *is on the local reigon boundary* **then**
    Extend the local region (dimension-wise);
    $k \leftarrow k + 1$;
  else
    $k \leftarrow k_{max} + 1$;
while $\mathbf{O}_s$ *belongs to* $\mathbf{A}$ **do**
  $\mathbf{O} \leftarrow$ Generate candidate offspring with $\mathbf{D_T}$;
  if *rand* $<$ *0.5* **then**
    $\mathbf{O}_s \leftarrow$ 1-sparsest solution in better-f $\mathbf{O}$;
  else
    $\mathbf{O}_s \leftarrow$ 1-sparsest solution in $\mathbf{O}$;
Evaluate $\mathbf{O}_s$ expensively;

In addition, to harness the potential of both promising infeasible solutions and feasible solutions, MGRLR conducts exploration in the converging region. This region consists of solutions with better-objective values than the current best-feasible solution, as well as solutions with equal or worse-objective values. By exploring the converging region, the algorithm can guide promising infeasible solutions toward the feasible region and steer solutions with worse-objective values toward regions with better-objective values, thereby facilitating convergence.

As presented in Algorithm 2, all the solutions that have been expensively evaluated (stored in an archive $\mathbf{A}$) are categorized into two sets. The solutions with better-objective values than the current best-feasible solution are moved to $\mathbf{A}_1$, while the remaining solutions are allocated to $\mathbf{A}_2$. The exploration types are labeled as feasible exploration, better-objective exploration, and constrained optimization (converging region), respectively. As shown in Algorithm 2 Lines 3–25, in accordance with the specific exploration type, the data for modeling, parents for offspring generation, and criteria for selecting solutions requiring expensive evaluation vary correspondingly.

The exploration of the feasible region aims to address the challenge of misguidance caused by the inaccurate surrogate model of the objective function. As shown in Algorithm 2 Lines 4–9, n_1 sparsest feasible solutions and n_2 sparsest infeasible solutions are used to construct the global surrogates (global surrogates are constructed within the original entire search space). In this study, the sparsity is measured by the minimum Euclidean distance. As shown in (9), $\mathbf{x}$ and $\mathbf{x}_j$ are two solutions in the design space. The sparsity of $\mathbf{x}$ in the archive is measured by the minimum distance of it to other solutions in the archive. When $\mathbf{x}$ comes from the candidate offspring, its sparsity is determined by the minimum distance of it to the solutions in the archive.

$$S(\mathbf{x}) = \min\left(\left\|\mathbf{x} - \mathbf{x}_j\right\|\right), \mathbf{x}_j \in \mathbf{A}. \tag{9}$$

If the number of feasible solutions is smaller than n_1, the solutions are sorted according to CV and the N_G best solutions are used to construct the global surrogates. This handling is not shown in Algorithm 2 to simplify the pseudo-code. Furthermore, the N sparsest feasible solutions are adopted as the parents for generating candidate offspring. If the feasible solutions are not enough, the N solutions with the smallest CV are chosen instead. Subsequently, the DE variants formulated in (4) and (5) are adopted to generate as diverse offspring as possible. Afterward, the offspring predicted to be feasible are sorted according to their sparsity. The offspring with the maximum sparsity is selected for expensive evaluation. If no offspring is predicted to be feasible, the offspring with the minimum predicted CV is chosen instead.

The exploration of the better-objective region serves to alleviate the misguiding issue arising from the inaccurate constraint surrogates. As shown in Algorithm 2 Lines 10–15, n_1 sparsest solutions with better-objective values than the current best-feasible solution and n_2 solutions with the best-objective values (but equal to or worse than the current best-feasible solution) are used for modeling. If the number of solutions with better-objective values is smaller than n_1, the N_G top-performed solutions regarding the objective are selected instead. Then, the N sparsest solutions with better-objective values are adopted as the parents. If the solutions with better-objective values are insufficient, the solutions are sorted by objective values and the best-N solutions are adopted instead. Then, the DE variants formulated in (4) and (5) are used to generate the candidate offspring. Afterward, the sparsest offspring predicted to possess better-objective values is selected for expensive evaluation. If no solution is predicted to have a better-objective value, the solution with the best-predicted objective value is selected instead.

The exploration of the converging region contributes to accelerating the convergence by leveraging both promising feasible and infeasible solutions. As presented in Algorithm 2 Lines 16–25, the solutions with n_1 smallest CV (among the solutions with better-objective values) and the n_2 top-performed solutions regarding the objective (among the solutions with equal or worse-objective values) are used to construct global surrogates. If the number of better solutions is inadequate, the N_G top-performed solutions regarding the objective are selected instead. By adopting both feasible and infeasible solutions for modeling, the surrogate accuracy around the feasible region boundary would be satisfactory. Similar operations are conducted to select N solutions as the parents. Subsequently, (5)–(6) are used to generate candidate offspring. To harness the information from both feasible and infeasible solutions simultaneously, the "best" in (6) contains

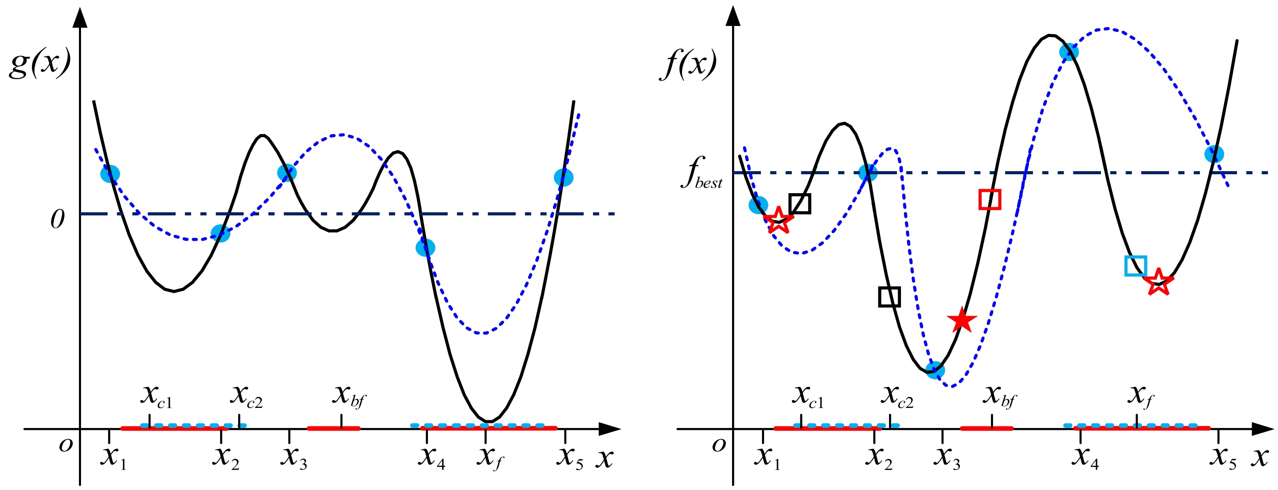

Fig. 1. Artificial minimization problem used to demonstrate the exploration process. The true constraint and objective functions are shown by solid curves: $g(x), f(x)$. The approximated functions are shown by dashed curves. The true feasible region is marked by the solid lines and the predicted feasible region is marked by the dashed lines. The global feasible optimum is denoted by a filled pentagram. Two local feasible optima are denoted by hollow pentagrams. The filled circles are the solutions in the archive. The squares are the possible candidate offspring. f_{best} is the objective value of the current best-feasible solution.

two solutions, namely, the best-feasible solution and the infeasible solution, possessing a better-objective value and the minimum CV. Moreover, to balance the objective optimization and the feasibility satisfaction, the solution chosen for expensive evaluation is determined as follows: if no solution in the archive or the set of candidate offspring has a better-objective value than the current best-feasible solution, the offspring possessing the predicted best-objective value is selected for expensive evaluation. Otherwise, FR is adopted to compare the offspring and the best one is chosen for expensive evaluation.

Remark 1: In this work, the uncertainty of a candidate offspring is simply measured by its sparsity defined based on Euclidean distance. However, other techniques can be utilized to assess the uncertainty of a solution. For instance, the Bayesian regression method can be employed to calculate the variance of the surrogate models [10], [22]. The variance can then serve as a measure of uncertainty for the solutions.

To further illustrate the exploration process, a 1-D artificial problem is presented. As shown in Fig. 1, a total of 5 solutions, denoted as x_1- x_5, have been expensively evaluated. Global surrogates are constructed using all the solutions and the population size is 2. Within the set of solutions, x_2 and x_4 are the two feasible solutions. x_2 is the current best-feasible solution and its objective value is f_{best}. The remaining 3 solutions are infeasible. x_1 and x_3 possess identical constraint function values and have better-objective values than f_{best}. x_5 is infeasible and its objective value is worse than f_{best}. It can be observed that the global feasible optimum lies in a small feasible region. However, the constraint surrogate fails to capture this feasible region.

In the feasible region exploration step of MGRLR, x_2 and x_4 are utilized to generate candidate offspring and the sparsest solution predicted to be feasible, e.g., x_f, is selected for expensive evaluation. Consequently, the surrogate of the objective in the feasible region can be improved. In the better-objective region exploration stage, x_1 and x_3 are employed to generate candidate offspring and the sparsest solution predicted to have a better-objective value, e.g., x_{bf}, is chosen for expensive evaluation. Consequently, the surrogate of the constraint can be improved. It can be observed that the feasible region containing the global feasible optimum is identified in this step. In the converging region exploration step, x_1 or x_3, along with x_2 is used to generate candidate offspring. Because there exist infeasible solutions with better-objective values, FR is used to select a solution for expensive evaluation. Therefore, x_{c2} is selected (predicted to be feasible). Although x_{c2} is determined to be infeasible after true evaluation. It contributes to improving surrogates around the global feasible optimum. As the iterations progress, it is expected that the MGRLR can guide the search toward the vicinity of the global feasible optimum.

In this study, the allocation of the computational budget to the three types of exploration is controlled by a parameter w. Within each generation, $2 \times w \times$ NFEs are used for the first two exploration types, while the remaining resources are used for exploring the converging region.

C. Local Exploitation

The local exploitation serves to enhance convergence. To achieve this goal, surrogate-based search methods are employed in MGRLR due to their high-sampling efficiency [10], [14].

As shown in Algorithm 3, for a solution $\mathbf{x}$, the N_L nearest solutions in the archive are used to construct the local surrogates. The lower and upper bounds of the initial local region are determined as the minimum and maximum values of the decision variables of these selected solutions, respectively.

After constructing the local surrogates, the solutions in the training set with better-objective values than f_{best} are counted. If there is at least one solution with a better-objective value, or if no feasible solution has been found so far, a constrained surrogate-based local search is performed. Otherwise, an unconstrained surrogate-based local search is

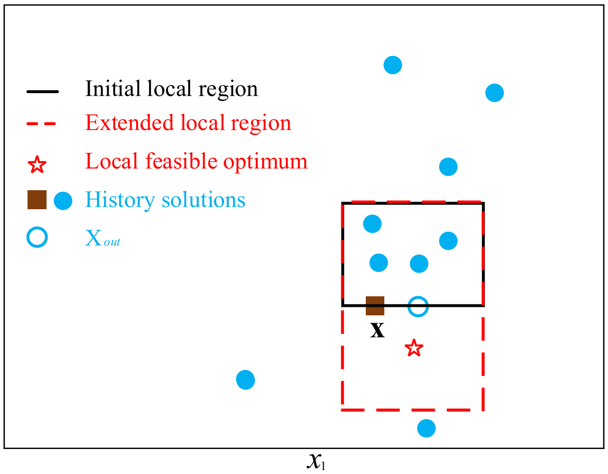

Fig. 2. Diagram showing the local region adaptation. Ten history solutions exist in the two-dimension design space. $\mathbf{x}$ is the location for local search. The five nearest solutions are used for constructing local surrogate models. $\mathbf{x}_{\text{out}}$ is the output solution of the local search within the initial local region.

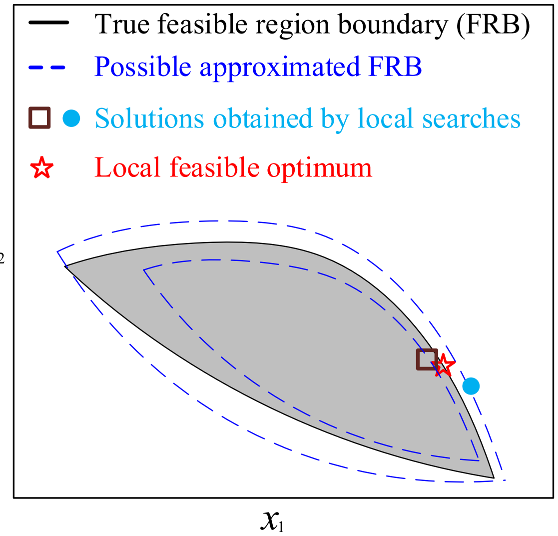

Fig. 3. Diagram showing the reward search. The filled circle denotes the solution obtained by the immediately previous local search. The hollow square denotes the solution obtained by the reward local search.

carried out (Algorithm 3 Lines 5–10). Furthermore, due to the restriction of the local region, the output of the local search may be at the region boundary. A possible situation is shown in Fig. 2, where the initial local region does not contain the local optimum. After the search, the output is on the local region boundary (the second dimension). Then the local region is enlarged by extending the second dimension (Algorithm 3 lines 11–15). In this way, the local region is more likely to contain a better solution. It should be noted that the search can also be led to better regions by constrained optimization in the global exploration stage. However, the extension strategy would be more efficient than the global search because the local surrogates are more accurate than the global surrogates in the local region. In addition, the maximum number of extensions of the local region is set to one. The reasons for this are as follows. First, the accuracy of the surrogates may decrease as the local region becomes larger. Second, trying to extend the local region too many times and performing local searches may become time-consuming. Therefore, we only try to extend the initial region once.

To determine whether the boundary is reached, a parameter $\text{LR}A_1$ is introduced. As formulated in

$$\begin{aligned}\mathbf{T}_{\text{dist}} &= (\mathbf{x}_{\max} - \mathbf{x}_{\min}) \times \text{LRA}_1\\ \mathbf{d}_L &= \arg_{\mathbf{d}}(\text{abs}(\mathbf{x}_{\min} - \mathbf{x}_{\text{out}}) < \mathbf{T}_{\text{dist}})\\ \mathbf{d}_U &= \arg_{\mathbf{d}}(\text{abs}(\mathbf{x}_{\max} - \mathbf{x}_{\text{out}}) < \mathbf{T}_{\text{dist}})\\ \mathbf{d} &= (1, 2, \ldots, D).\end{aligned} \tag{10}$$

$\mathbf{T}_{\text{dist}}$ is the allowed boundary tolerance for not enlarging the local region. If there exist values of $\mathbf{x}_{\text{out}}$ close to the boundary, the corresponding dimension $\mathbf{d}_L$ or $\mathbf{d}_U$ is extended as in

$$\begin{aligned}\mathbf{x}_r &= \mathbf{x}_{\max} - \mathbf{x}_{\min}\\ \mathbf{x}_{\min}(\mathbf{d}_L) &= \max(\text{lb}(\mathbf{d}_L), \mathbf{x}_{\min}(\mathbf{d}_L) - \mathbf{x}_r(\mathbf{d}_L) \times \text{LRA}_2)\\ \mathbf{x}_{\max}(\mathbf{d}_U) &= \min(\text{ub}(\mathbf{d}_U), \mathbf{x}_{\max}(\mathbf{d}_U) + \mathbf{x}_r(\mathbf{d}_U) \times \text{LRA}_2)\end{aligned} \tag{11}$$

where lb and ub are, respectively, the lower and upper bounds of the entire search space. LRA_2 is a parameter that determines how much the local region is enlarged.

Due to the high-sampling efficiency of the local search, the resultant solution is likely to be a solution in the archive. In such cases, evaluating the most uncertain solution can enhance exploration within the local region. To achieve this goal, the solutions in the training set are employed to generate candidate offspring using (5)–(6). Then either the sparsest solution predicted to have a better-objective value, or the sparsest solution, is selected for expensive evaluation with equal probability (Algorithm 3 Lines 16–21). In the former case, the accuracy of the surrogates can be improved within the better-objective region. In the latter case, the surrogates are improved in a larger local region.

Although the local surrogates are relatively more accurate than the global surrogates, there are still approximation errors. Therefore, infeasible solutions with better-objective values may be obtained. As shown in Fig. 3, after the infeasible solution with a better-objective value is expensively evaluated, the accuracy of the surrogates around the optimum can be improved. Consequently, the reward constrained local search is expected to yield a solution that approaches the optimum from the feasible side.

To benefit from both promising feasible and infeasible solutions, two types of solutions are determined as the local search locations. First, the solutions with better-objective values than the current best-feasible solution are sorted by CV. Second, the solutions with equal or worse-objective values are compared based on the objective values. Then, the best solutions from the two types of solutions are selected as local search locations. In this study, the computational resources for local search are equally allocated to the two types of locations.

Remark 2: In MGRLR, local regions are adjusted based on the search information from surrogates, without expensive evaluations. Moreover, infeasible solutions with better-objective values are given high emphasis, which distinguishes MGRLR from the method proposed in reference [22], [23].

IV. Experimental Study

In this section, an artificial problem is solved by MGRLR to demonstrate the search process. The experimental study is then carried out. Specifically, the test problems and 4 state-of-the-art algorithms for comparison are first briefly introduced. Then, the experimental settings are given. This is followed by the component and parameter analysis of the proposed MGRLR. Finally, the comparison results are presented.

A. Example Illustration

Due to the page limitation, the example problem and the search process of MGRLR are illustrated in the supplementary file. It can be observed that the proposed strategies work as intended. Consequently, the global feasible optimum of the example problem can be successfully found by MGRLR.

B. Problems and Algorithms

Three widely used test suites are employed to study the effectiveness of the proposed MGRLR, i.e., CEC2006 [54], CEC2010 [55], and CEC2017 [56]. In addition, only problems with inequality constraints are considered. In this study, we assume that the evaluation of a solution is expensive. The algorithms adopted for comparison are listed as follows.

1) *GloSADE [10]:* A global exploration and local exploitation framework is presented in this algorithm. In the exploration stage, global surrogates are constructed with all the history solutions, and two variants of DE are employed to generate candidate offspring based on the population. In addition, FR is used for environmental selection and model management (strategies for selecting offspring for expensive evaluations). In the local exploitation stage, the individuals in the population serve as the locations for performing surrogate-based local searches.
2) *SACCDE [11]:* In this method, global surrogates are constructed using all the historical solutions. Then, the candidate offspring are generated by exploiting the information of the best solution under FR. Furthermore, a classification-collaboration mechanism is incorporated to provide promising mutation vectors. The FR is modified and applied in the environment selection step. In addition, the modified FR is used to select the best candidate offspring for expensive evaluation.
3) *MPMLS [20]:* This method uses only local surrogates. The local surrogates are constructed around each individual in the population. A local search region is defined around each individual and the entire search space is covered. In addition, a penalty method with adaptive adjustment of penalty coefficients is proposed to guide the environmental selection and model management.
4) *DSI [17]:* This method selects data of interest for the construction of global surrogates. Promising solutions based on FR and uncertain solutions are adaptively chosen for expensive evaluation. In addition, this framework belongs to the evolution control methods [5], together with the development of a search intensity adjustment strategy.

C. Experiment Settings

For all algorithms compared, the maximum number of function evaluations is set to 1000. For each algorithm, 25 independent runs are performed on each test problem. For the CEC2006 problems, the global feasible optima of the test problems have been reported in [54]. Therefore, the function error (the difference between the reported optimum and the best-feasible objective value found by an algorithm) is used to compare the performance. For problems in CEC2010 and CEC2017, the objective value of the obtained best-feasible solution is recorded. If an algorithm fails to find a feasible solution in at least one run, the number of runs in which at least a feasible solution is successfully found is reported.

The parameter settings of GloSADE, SACCDE, MPMLS, and DSI are the same as in the original papers, except for the maximum number of function evaluations. Moreover, the DSI is embedded in the Constrained Composite DE (C^2oDE) [57] for comparison.

For the proposed MGRLR, sequential quadratic programming (SQP) [28] is used as the search engine for surrogate-based local search. The main parameters of MGRLR are given in Table I. In addition, the scaling factor F and the crossover probability CR in DE are randomly chosen from {0.2:0.2:1} and {0.4:0.2:1}, respectively.

The Wilcoxon rank-sum test and the Friedman test with a significance level of 0.05 are performed to analyze the results statistically. The Hommel post-hoc procedure is adopted to adjust the p-value of the Friedman test [58].

D. Component Analysis

First, the effectiveness of each component in MGRLR is verified on 13 problems in CEC2006. The variants are listed in Table II.

The detailed results are presented in Table S-I in the supplementary file. The summarized results are presented in Table III. The symbols "+," "−," and "=" denote the number of problems on which the compared algorithm is significantly better than, worse than, and competitive with MGRLR in the Wilcoxon rank-sum test. For brevity, these symbols have the same meaning in the following tables. The best results are highlighted with gray background.

It can be observed that the MGRLR obtains the best-average rank among the variants. The comparison results with MGRLR_FR suggest that MGRLR can benefit from the promising infeasible solutions, especially for problems on which the global feasible optimum lies on the boundary of the feasible region, e.g., G9 and G10. MGRLR shows better performance on such problems because the feasible solutions are led to regions with better-objective values, while at the same time, the promising infeasible solutions are led to feasible regions. For problems where the global feasible

TABLE I
PARAMETER SETTINGS IN MGRLR

Parameter	Value
N: population size	min $(5 \times D, 60)$
w: $w \times N$ *FEs* for feasible region exploration and $w \times N$ *FEs* for better-objective region exploration in each generation	0.2
N_G: number of solutions for constructing global surrogate models	200
N_L: number of solutions for constructing local surrogate models	min $(5 \times D, 100)$
(LRA_1, LRA_2): parameters that control the adjustment process of local regions	(0.1, 0.5)
q: number of iterations of SQP	300
NPr: number of candidate offspring generated by each DE variant	500

TABLE II
VARIANTS OF MGRLR

Variants	
MGRLR_FR	Use FR in the following steps: select solutions for constructing global surrogate models and expensive evaluation in the converging region exploration stage; determine the local search locations
MGRLR_NoF	Remove the feasible region exploration stage
MGRLR_NoB	Remove the better-objective region exploration stage
MGRLR_NoC	Remove the converging region exploration stage (*w* = 0.5)
MGRLR_NoG	Remove the entire global exploration stage
MGRLR_NoR	Remove the reward search mechanism
MGRLR_SL	Remove the local region adaptation strategy (static local region)

TABLE III
COMPARISON RESULTS OF THE VARIANTS OF MGRLR

	MGRLR_FR	MGRLR_NoF	MGRLR_NoB	MGRLR_NoC
+/-/=	1/9/3	0/6/7	1/6/6	1/6/6
Rank	5.92	4.42	4.65	4.81
p-value	0.005	0.843	0.441	0.278
	MGRLR_NoG	MGRLR_NoR	MGRLR_SL	MGRLR
+/-/=	5/4/4	0/5/8	0/8/5	NA
Rank	2.92	5.23	5.65	2.38
p-value	1	0.069	0.014	NA

optimum is far from the feasible region boundary, e.g., G8, MGRLR_FR is better than MGRLR because FR prefers feasible solutions more. As a result, the solutions for generating candidate offspring in the converging region exploration step and the locations for local search in MGRLR_FR are closer to the global feasible optimum than MGRLR. Therefore, MGRLR_FR can generate more solutions around the global feasible optimum than MGRLR on such problems.

Compared with MGRLR_NoF, it can be observed that MGRLR is much better than MGRLR_NoF on G8. Since there are multiple local optima in the feasible region, MGRLR_NoF is easily trapped in some local regions. For problems on which the global feasible optimum lies on the boundary of a large feasible region (relative to other feasible regions), MGRLR and MGRLR_NoF perform similarly, e.g., G4.

In comparison with MGRLR_NoB, MGRLR performs better on 6, worse on 1, and similar on 6 out of the 13 problems. This suggests that the difficulties caused by inaccurate constraint surrogate models can be effectively mitigated by exploring the better-objective region in MGRLR. For instance, due to the disconnected feasible regions, the search of MGRLR_NoB on G12 is easy to get trapped in a local feasible region. In contrast, MGRLR can find the global feasible optimum if enough resources are allocated to the better-objective exploration stage.

Compared with MGRLR_NoC, MGRLR performs better on 6, worse on 1, and similar on 6 problems. The results indicate that exploring the converging region can facilitate convergence. However, the local search appears to be more efficient on some other problems, e.g., G10. The objective function and 3 of the 6 constraint functions of G10 are linear. Therefore, the surrogate accuracy can be improved quickly. As a result, local search can be more efficient than the surrogate-assisted prescreening strategies in the converging region exploration stage.

MGRLR_NoG performs better than MGRLR on 5, worse on 4, and competitive on 4 problems. In particular, although MGRLR and MGRLR_NoG find similar solutions on G7, G9, G10, and G19, the accuracy of the best-feasible solution obtained by MGRLR is worse. This is because MGRLR_NoG allocates more resources to local exploitation, and the exploration strategies seem unnecessary for these problems. However, MGRLR_NoG fails to find a feasible solution in one run on G6 and gets trapped in a local search region on G8 and G12. This suggests that the exploration strategies make MGRLR more robust and capable of handling different types of problems.

The comparison results between MGRLR_NoR and MGRLR indicate that the proposed reward search mechanism improves convergence effectively. As illustrated in Fig. 3, there is a high probability of obtaining a better-feasible solution if a reward search is carried out in time. In addition, the comparison results between MGRLR_SL and MGRLR suggest that the strategies to extend the local region prove to be advantageous in further enhancing convergence.

In addition, to study the influence of the strategy used to select the infeasible solutions with better-objective values, four kinds of methods are implemented within MGRLR, respectively. The details of these methods and the experimental results are given in the supplementary file Section III-B. The results suggest that selecting the infeasible solutions based on CV values only may strike a better balance between exploration and exploitation on most problems. In the early search phase, solutions with similar CV ranks may come from different regions of the search space. Hence, the infeasible solutions may exhibit good diversity, allowing for exploration in various regions. As the search progresses, a growing number of solutions with smaller CV values in each local region can be discovered. Consequently, the selected infeasible solutions are expected to converge toward the solution with the smallest CV value, particularly when the global feasible optimum is located on the boundary of the feasible region. In such instances, the search tends to exploit the local region that contains the solution with the smallest CV value. On the contrary, maintaining diversity through clustering methods can lead to slower convergence in the late search phase of MGRLR.

The influence of the surrogate models is also investigated by training the surrogate models with four kinds of methods. The details of these methods and the experimental results are given in the supplementary file Section III-C. The results demonstrate that the RBF with a cubic basis function and a linear polynomial tail gives the best performance of MGRLR on most problems. It is worth noting that this surrogate technique has also been employed in [10], [11], [20], and [33].

In summary, the results suggest that MGRLR can strike a good balance between exploration and exploitation.

E. Parameter Analysis

The parameters N, w, N_G, $(\mathrm{LRA}_1, \mathrm{LRA}_2)$, q, NPr, and N_L in MGRLR are empirically studied using the 13 problems in CEC2006. During the study of each parameter, the remaining parameters are fixed as specified in Table I. The detailed results can be found in Tables S-IV–S-XI.

In this study, $N = \min(5D, 60)$ is used as the population size. To analyze the impact of N, the latter value in the parenthesis is varied as 20, 40, 60, 80, and 100, respectively, for comparison. It can be observed that $N = \min(5D, 60)$ obtains the best-average rank. When $N = \min(5D, 20)$, MGRLR deteriorates on some problems, e.g., G1 ($D = 13$) and G7 ($D = 7$). Moreover, when $N = \min(5D, 100)$, the performance of MGRLR degrades on G19 ($D = 15$). In addition, $N = \min(5D, 40)$, $N = \min(5D, 60)$, and $N = \min(5D, 80)$ yield similar results across most problems. Therefore, $N = \min(5D, 60)$ is recommended.

For $w = 0.1, 0.2, 0.3$, the setting $w = 0.2$ obtains the best-average rank. In addition, for $w = 0.1$, the exploration may be insufficient on some problems. For instance, on G9, G10 and G12, w = 0.1 obtains the worst results. For $w = 0.3$, the exploitation may be insufficient. For instance, the setting $w = 0.3$ gives rise to the worst results on G4. However, it should be pointed out that the difference between $w = 0.2$ and w = 0.3 is not significant.

The allowed maximum number of solutions for constructing global surrogates, i.e., N_G, is varied as 50, 100, 200, 300, 400, 600, 800, and 1000, respectively, to study its impact. As shown in Fig. 4, the computation time increases as the number of solutions increases. However, $N_G = 200$ gives the best-average rank. Thus, this setting is recommended. The worse performance when $N_G < 200$ may be caused by inaccurate surrogates. Due to the misguiding issue, it is difficult to identify high-quality solutions for expensive evaluation. For $N_G > 200$, the surrogate accuracy may decrease [13].

For the local region adaptation parameters, i.e., (LRA_1, LRA_2), LRA_1 is used to determine whether the local region should be extended (determine $\mathbf{T}_{\text{dist}}$). The smaller the LRA_1 is, the less frequently the local region will be extended. LRA_2 is used to determine how much to extend the local region. If it is too small, the local region cannot be extended effectively. In contrast, if the local region is too large, the local search may lose its exploitation capability. As shown in Fig. 5, the combination of (0.1, 0.5) obtains the best-average rank. Therefore, this setting is recommended.

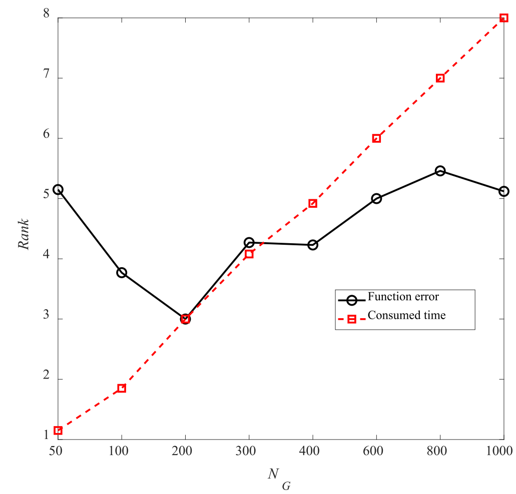

Fig. 4. Average ranks of MGRLR with different N_G in terms of function errors and consumed time (s) on 13 problems in CEC2006.

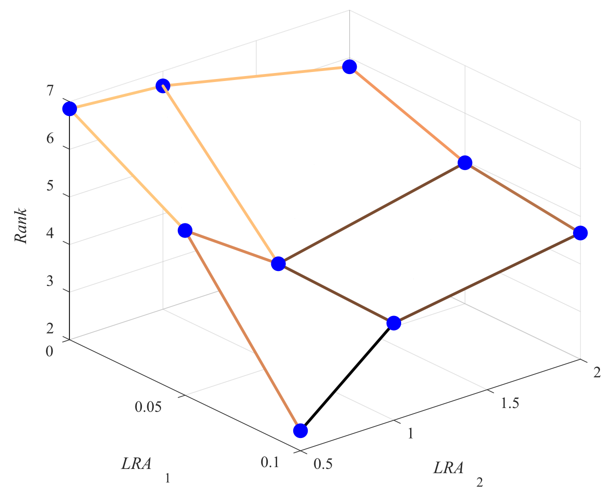

Fig. 5. Average ranks of MGRLR with different $(\mathrm{LRA}_1, \mathrm{LRA}_2)$ in terms of function errors on 13 problems in CEC2006.

The number of iterations (q) in SQP affects the strength of the local search. The results demonstrate that for most problems there is no significant difference in performance when this parameter is varied from 100 to 500 (step = 100).

The effect of the number of candidate offspring (NPr) generated by each DE variant is studied by varying it from 100 to 500 (step = 100). The results suggest that the average performance of MGRLR is improved progressively as NPr increases.

The N_L parameter determines the number of solutions used to construct local surrogate models ($\min(5D, 100)$). The former value in the parenthesis is set to $1D$, $2D$, $3D$, $4D$, $5D$, $6D$, and $7D$ to study its effect. The results suggest that the variation in performance is not significant when this parameter is increased from $5D$ to $7D$. Considering the computational complexity, $5D$ is recommended for this parameter.

F. Comparison With State-of-the-Art Algorithms

The detailed comparison results of MGRLR with four algorithms are given in Tables S-XII–S-XVII and

TABLE IV
COMPARISON RESULTS OF GLOSADE, SACCDE, MPMLS, AND MGRLR ON CEC2006

	GloSADE	SACCDE	MPMLS	DSI	MGRLR
+/-/=	0/13/0	1/12/0	1/11/1	0/13/0	NA
Rank	3.92	2.62	3.5	2.92	1.04
p-value	3.16E-05	3.16E-04	6.99E-04	0.023	NA

TABLE V
COMPARISON RESULTS OF GLOSADE, SACCDE, MPMLS, AND MGRLR ON CEC2010

$D = 10$	GloSADE	SACCDE	MPMLS	DSI	MGRLR
+/-/=	0/6/0	0/4/2	0/4/2	0/5/1	NA
Rank	4.25	4.08	2	3.17	1.5
p-value	0.025	0.045	1	0.667	NA
$D = 30$	GloSADE	SACCDE	MPMLS	DSI	MGRLR
+/-/=	0/5/1	2/4/0	0/4/2	0/3/3	NA
Rank	3.5	3.58	3.25	2.83	1.83
p-value	0.667	0.542	1	1	NA

TABLE VI
COMPARISON RESULTS OF GLOSADE, SACCDE, MPMLS, AND MGRLR ON CEC2017

$D = 10$	GloSADE	SACCDE	MPMLS	DSI	MGRLR
+/-/=	0/7/2	0/6/3	0/6/3	2/3/4	NA
Rank	3.89	3.44	3.33	2.67	1.67
p-value	0.007	0.068	0.112	1	NA
$D = 30$	GloSADE	SACCDE	MPMLS	DSI	MGRLR
+/-/=	0/5/4	2/1/6	0/6/3	2/4/3	NA
Rank	3.83	2.61	3.67	2.78	2.11
p-value	0.038	1	0.090	1	NA

Figs. S-IV–S-VIII, respectively. The summary of the comparison results in terms of the objective optimization is given in Tables IV–VI.

The proposed MGRLR demonstrates the best-average ranks across all three test suites. In addition, it consumes the second least computational time, with MPMLS being the most efficient in terms of computation time. The reason for this is that MPMLS utilizes only local surrogates, while global surrogates are used in the other four methods. The difference among the four methods lies in how the global surrogates are utilized. To be specific, GloSADE updates the global surrogates in each generation, resulting in less frequent updates. This delays the use of the information of the newly added solutions. In contrast, SACCDE and MGRLR update the global surrogates as soon as new solutions are added. However, the updating process would become time-consuming if all solutions are used. In MGRLR, only a subset of the solutions is selected for modeling, depending on the type of exploration. Although DSI selects solutions of interest for modeling, most of the solutions are used. Furthermore, DSI involves an iterative search method embedded within it, which is more time-consuming than surrogate-assisted prescreening methods. As a result, MGRLR is more computationally efficient than GloSADE, SACCDE, and DSI.

In terms of the objective optimization results, MGRLR achieves the best results on 13, 12, 11, and 13 out of the 13 problems in CEC2006 compared to GloSADE, SACCDE, MPMLS, and DSI, respectively. In particular, MGRLR, SACCDE, MPMLS, and DSI achieve faster convergence than GloSADE on most problems because the first four methods update surrogates more frequently than GloSADE. Furthermore, MGRLR and DSI demonstrate faster convergence compared to SACCDE and MPMLS on most problems. This is because MGRLR utilizes iterative surrogate-based local searches, while DSI involves an iterative optimization process before selecting solutions for expensive evaluation. As a result, these two methods achieve better convergence. However, although DSI incorporates uncertainty-oriented model management strategies, the solution for expensive evaluation is selected from the population. As a result, DSI may get trapped in a local search region when the population loses diversity, e.g., G8 and G12. In addition, SACCDE shows promising convergence as it frequently uses the best solution to generate candidate offspring. However, it may fail on problems with multiple models or disconnected feasible regions, e.g., G8 and G12. MPMLS maintains diversity by selecting solutions from multiple local regions for expensive evaluation. Therefore, it has a relatively stronger exploration capability compared to DSI, SACCDE, and GloSADE, e.g., G02. However, excessive exploration in MPMLS can lead to inadequate exploitation, e.g., G4, G7, and G9. In contrast, MGRLR converges quickly without significant degradation of exploration ability on most problems, except for G18. The reason may be that the better-objective region is large, but the feasible region is small, causing MGRLR to struggle in finding the feasible region containing the optimum.

For CEC2010, MGRLR is better than GloSADE, SACCDE, MPMLS, and DSI on 6, 4, 4, and 5 out of the 6 10D problems, respectively. In the 30D situation, the numbers are 5, 4, 4, and 3, respectively. In particular, the MGRLR outperforms the other four algorithms on G7, G8, and G14 (10D and 30D). On G15 (10D), MGRLR successfully finds feasible solutions while the others fail. However, all the compared algorithms fail to find a feasible solution on G15 (30D). In addition, MGRLR performs similarly to MPMLS on G1 and G13 (10D), with MPMLS yielding the best results.

For 10D problems in CEC2017, MGRLR achieves the best performance on G4, G13, G20, and G22. Although MGRLR is outperformed by DSI on G1 and G2, it achieves the second-best performance on these two problems. In addition, DSI fails to find feasible solutions on G13 and G22 in 1 and 5 runs, respectively. Moreover, SACCDE performs similarly to MGRLR on G20. MPMLS performs similarly to MGRLR on G13. For 30D problems, MGRLR outperforms the compared algorithms on G5. DSI performs best on G1 and G2, with MGRLR performing third best on G1 and second best on G2. In addition, MGRLR is competitive with GloSADE and SACCDE on G20. SACCDE and MGRLR perform similarly on G2. On G1 and G4, MGRLR is beaten by SACCDE. However, MGRLR is the second-best performer on G4.

In summary, MGRLR demonstrates promising performance on the test problems, particularly when the global feasible optimum is close to the feasible region boundary.

V. Conclusion

In this article, a novel SAEA for expensive constrained optimization problems with inequality constraints is proposed. In the proposed method, the exploration is performed in three kinds of global regions, i.e., the feasible region, the better-objective region, and the converging region, sequentially in each generation. To improve efficiency, the solutions for modeling, the parents for offspring generation, and the solutions selected for expensive evaluation are adaptively determined according to the exploration types. In addition, to facilitate convergence, two types of local regions are further searched, i.e., the local regions with better-objective values (sorted by constraint violations) and the local regions with worse-objective values (sorted by objective values). Furthermore, a reward local search is performed when the local search discovers an infeasible solution with a better-objective value.

The effectiveness of the proposed strategies in MGRLR is empirically verified on 13 problems in the CEC2006 test suite through the comparison of the different variants of MGRLR. The results suggest that: 1) exploration in feasible regions can effectively lead the search out of local optima; 2) exploration in better-objective regions can mitigate the risks of being trapped in local feasible regions when problems with disconnected feasible regions are encountered; 3) exploration in the converging region enables the search to converge to the optimum quickly; and 4) local search in the two types of local regions can further facilitate convergence by adaptively adjusting the local regions and performing the reward search. In addition, the recommended parameter settings are provided in Table I.

In this study, MGRLR is compared with four state-of-the-art algorithms for ECOPs on problems collected from CEC2006, CEC2010 (D = 10, 30), and CEC2017 (D = 10, 30). The results suggest that MGRLR can effectively reduce the computation time and achieve better or competitive performance on most of the tested problems. In particular, MGRLR is suitable for problems in which the optimal solution lies on the boundary of the feasible region. However, if the optimum is distant from the boundary of the feasible region and the feasible region is large, the efficacy of MGRLR may reduce. Furthermore, addressing problems that have large infeasible better-objective regions and disconnected feasible regions remains a challenging task.

The proposed MGRLR assumes that an evaluation yields the values of both the objective and constraint functions simultaneously. Nevertheless, in numerous engineering applications, the objective and constraint functions are evaluated independently [12], [43]. Moreover, in certain scenarios, it becomes even more challenging as the infeasible solutions cannot be evaluated. Thus, extending MGRLR to address such problems will be part of our future work. Furthermore, the generalization of the concepts utilized in MGRLR for addressing expensive constrained multiobjective optimization problems is an intriguing and worthwhile direction.

References

[1] Z. Yang, H. Qiu, L. Gao, L. Chen, and X. Cai, “Constraint boundary pursuing-based surrogate-assisted differential evolution for expensive optimization problems with mixed constraints,” *Struct. Multidiscip. Optim.*, vol. 66, no. 2, p. 40, 2023, doi: 10.1007/s00158-022-03473-w.

[2] Y. Liu, J. Liu, Y. Jin, F. Li, and T. Zheng, “A surrogate-assisted two-stage differential evolution for expensive constrained optimization,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 715–730, Jun. 2023, doi: 10.1109/tetci.2023.3240221.

[3] Y. Jin, H. Wang, and C. Sun, *Data-Driven Evolutionary Optimization*. Berlin, Germany: Springer, 2021.

[4] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*. Hoboken, NJ, USA: Wiley, 2008.

[5] Y. Jin, M. Olhofer, and B. Sendhoff, “A framework for evolutionary optimization with approximate fitness functions,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 5, pp. 481–494, Oct. 2002.

[6] E. Krempser, H. S. Bernardino, H. J. Barbosa, and A. C. Lemonge, “Differential evolution assisted by surrogate models for structural optimization problems,” in *Proc. Int. Conf. Comput. Struct. Technol. (CST)*, 2012, pp. 1–19.

[7] D. Buche, N. N. Schraudolph, and P. Koumoutsakos, “Accelerating evolutionary algorithms with Gaussian process fitness function models,” *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 2, pp. 183–194, May 2005.

[8] C. K. Goh, D. Lim, L. Ma, Y. S. Ong, and P. S. Dutta, “A surrogate-assisted memetic co-evolutionary algorithm for expensive constrained optimization problems,” in *Proc. IEEE Congr. Evol. Comput.*, 2011, pp. 744–749.

[9] B. Liu, Q. Zhang, and G. G. E. Gielen, “A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.

[10] Y. Wang, D.-Q. Yin, S. Yang, and G. Sun, “Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints,” *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1642–1656, May 2019.

[11] Y. Zan, Q. Haobo, G. Liang, C. Xiwen, J. Chen, C. Liming, “Surrogate-assisted classification-collaboration differential evolution for expensive constrained optimization problems,” *Inf. Sci.*, vol. 508, pp. 50–63, Jan. 2020, doi: 10.1016/j.ins.2019.08.054.

[12] K. H. Rahi, H. K. Singh, and T. Ray, “Partial evaluation strategies for expensive evolutionary constrained optimization,” *IEEE Trans. Evol. Comput.*, vol. 25, no. 6, pp. 1103–1117, Dec. 2021.

[13] Z. Song, H. Wang, C. He, and Y. Jin, “A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 25, no. 6, pp. 1013–1027, Dec. 2021, doi: 10.1109/TEVC.2021.3073648.

[14] P. Jiang, Y. Cheng, J. Yi, and J. Liu, “An efficient constrained global optimization algorithm with a clustering-assisted multiobjective infill criterion using Gaussian process regression for expensive problems,” *Inf. Sci.*, vol. 569, pp. 728–745, Aug. 2021, doi: 10.1016/j.ins.2021.05.015.

[15] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, “An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems,” *Inf. Sci.*, vol. 561, pp. 304–325, Jun. 2021, doi: 10.1016/j.ins.2020.11.056.

[16] H. Akbari and A. Kazerooni, “KASRA: A Kriging-based adaptive space reduction algorithm for global optimization of computationally expensive black-box constrained problems,” *Appl. Soft. Comput.*, vol. 90, May 2020, Art. no. 106154, doi: 10.1016/j.asoc.2020.106154.

[17] Z. Song, H. Wang, and Y. Jin, “A surrogate-assisted evolutionary framework with regions of interests-based data selection for expensive constrained optimization,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 10, pp. 6268–6280, Oct. 2023, doi: 10.1109/tsmc.2023.3281822.

[18] S. D. Handoko, C. K. Kwoh, and Y.-S. Ong, “Feasibility structure modeling: An effective chaperone for constrained memetic algorithms,” *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 740–758, Oct. 2010.

[19] R. G. Regis, “Trust regions in surrogate-assisted evolutionary programming for constrained expensive black-box optimization,” in *Evolutionary Constrained Optimization*. New York, NY, USA: Springer, 2015, pp. 51–94.

[20] G. Li and Q. Zhang, “Multiple penalties and multiple local surrogates for expensive constrained optimization,” *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 769–778, Aug. 2021, doi: 10.1109/TEVC.2021.3066606.

[21] C. Liu, Z. Wan, Y. Liu, X. Li, and D. Liu, "Trust-region based adaptive radial basis function algorithm for global optimization of expensive constrained black-box problems," *Appl. Soft. Comput.*, vol. 105, Jul. 2021, Art. no. 107233, doi: 10.1016/j.asoc.2021.107233.

[22] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek, "Scalable global optimization via local Bayesian optimization," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 1–12, Oct. 2019.

[23] D. Eriksson and M. Poloczek, "Scalable constrained Bayesian optimization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 730–738.

[24] R. G. Regis, "A survey of surrogate approaches for expensive constrained black-box optimization," in *Proc. World Congr. Global Optim.*, 2019, pp. 37–47.

[25] K. R. Opara and J. Arabas, "Differential evolution: A survey of theoretical analyses," *Swarm Evol. Comput.*, vol. 44, pp. 546–558, Feb. 2019, doi: 10.1016/j.swevo.2018.06.010.

[26] R. Agrawal, K. Deb, and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–48, 2000.

[27] M.-E. Miranda-Varela and E. Mezura-Montes, "Surrogate-assisted differential evolution with an adaptive evolution control based on feasibility to solve constrained optimization problems," in *Proc. Fifth Int. Conf. Soft Comput. Problem Sol.*, 2016, pp. 809–822.

[28] C. T. Lawrence and A. L. Tits, "A computationally efficient feasible sequential quadratic programming algorithm," *SIAM J. Optim.*, vol. 11, no. 4, pp. 1092–1118, 2001.

[29] R. G. Regis "A hybrid between a surrogate-assisted evolutionary algorithm and a trust region method for constrained optimization," in *Proc. Genetic Evol. Comput. Conf. Comp.*, 2019, pp. 324–325.

[30] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, 2000, doi: 10.1016/S0045-78259900389-8.

[31] Z. Yang, A. Kumar, R. Mallipeddi, and D.-G. Lee, "Spherical search with epsilon constraint and gradient-based repair framework for constrained optimization," *Swarm Evol. Comput.*, vol. 82, Oct. 2023, Art. no. 101370, doi: 10.1016/j.swevo.2023.101370.

[32] Z. Yang, H. Qiu, L. Gao, C. Jiang, L. Chen, and X. Cai, "A novel surrogate-assisted differential evolution for expensive optimization problems with both equality and inequality constraints," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2019, pp. 1688–1695.

[33] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 326–347, Jun. 2014.

[34] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 61–70, 2011.

[35] Y. Zeng, Y. Cheng, and J. Liu, "An efficient global optimization algorithm for expensive constrained black-box problems by reducing candidate infilling region," *Inf. Sci.*, vol. 609, pp. 1641–1669, Sep. 2022, doi: 10.1016/j.ins.2022.07.162.

[36] M. Schonlau, W. J. Welch, and D. R. Jones, "Global versus local search in constrained optimization of computer models," in *Lecture Notes-Monograph Series*, vol. 34. Bengaluru: Inst. Math. Stat., 1998, pp. 11–25.

[37] J. M. Parr, A. J. Keane, A. I. J. Forrester, and C. M. E. Holden, "Infill sampling criteria for surrogate-based optimization with constraint handling," *Eng. Optim.*, vol. 44, no. 10, pp. 1147–1166, 2012. [Online]. Available: https://www.tandfonline.com/doi/pdf/10.1080/0305215X.2011.637556?needAccess=true

[38] R. Jiao, S. Zeng, C. Li, Y. Jiang, and Y. Jin, "A complete expected improvement criterion for Gaussian process assisted highly constrained expensive optimization," *Inf. Sci.*, vol. 471, pp. 80–96, Jan. 2019.

[39] H. Dong, P. Wang, C. Fu, and B. Song, "Kriging-assisted teaching-learning-based optimization (KTLBO) to solve computationally expensive constrained problems," *Inf. Sci.*, vol. 556, pp. 404–435, May 2021, doi: 10.1016/j.ins.2020.09.073.

[40] J. Liu, Y. Wang, G. Sun, and T. Pang, "Solving highly expensive optimization problems via evolutionary expected improvement," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 8, pp. 4843–4855, Aug. 2023, doi: 10.1109/tsmc.2023.3257030.

[41] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664–2677, Sep. 2017.

[42] J. Kůdela and R. Matoušek, "Combining Lipschitz and RBF surrogate models for high-dimensional computationally expensive problems," *Inf. Sci.*, vol. 619, pp. 457–477, Jan. 2023, doi: 10.1016/j.ins.2022.11.045.

[43] Y. Zhang, X.-F. Ji, X.-Z. Gao, D.-W. Gong, and X.-Y. Sun, "Objective-constraint mutual-guided surrogate-based particle swarm optimization for expensive constrained multimodal problems," *IEEE Trans. Evol. Comput.*, vol. 27, no. 4, pp. 908–922, Aug. 2023, doi: 10.1109/tevc.2022.3182810.

[44] G. Li, Q. Zhang, Q. Lin, and W. Gao, "A three-level radial basis function method for expensive optimization," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 5720–5731, Jul. 2022, doi: 10.1109/TCYB.2021.3061420.

[45] W. Yi, Y. Chen, Z. Pei, and J. Lu, "Adaptive differential evolution with ensembling operators for continuous optimization problems," *Swarm Evol. Comput.*, vol. 69, Mar. 2022, Art. no. 100994, doi: 10.1016/j.swevo.2021.100994.

[46] C. L. Camacho-Villalón, M. Dorigo, and T. Stützle, "PSO-X: A component-based framework for the automatic design of particle swarm optimization algorithms," *IEEE Trans. Evol. Comput.*, vol. 26, no. 3, pp. 402–416, Jun. 2022, doi: 10.1109/TEVC.2021.3102863.

[47] H.-G. Beyer and B. Sendhoff, "Simplify your covariance matrix adaptation evolution strategy," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 746–759, Oct. 2017.

[48] K. Qiao, J. Liang, K. Yu, M. Yuan, B. Qu, and C. Yue, "Self-adaptive resources allocation-based differential evolution for constrained evolutionary optimization," *Knowl. Based Syst.*, vol. 235, Jan. 2022, Art. no. 107653, doi: 10.1016/j.knosys.2021.107653.

[49] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Inf. Sci.*, vol. 423, pp. 172–186, Jan. 2018, doi: 10.1016/j.ins.2017.09.053.

[50] T. Kawachi, J.-I. Kushida, A. Hara, and T. Takahama, "Efficient constraint handling based on the adaptive penalty method with balancing the objective function value and the constraint violation," in *Proc. IEEE Int. Workshop Comput. Intell. Appl. (IWCIA)*, 2019, pp. 121–128, doi: 10.1109/IWCIA47330.2019.8955094. [Online]. Available: https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp

[51] C. A. C. Coello, "Treating constraints as objectives for single-objective evolutionary optimization," *Eng. Optim.*, vol. 32, no. 3, pp. 275–308, 2000.

[52] T. P. Runarsson, "Constrained evolutionary optimization by approximate ranking and surrogate models," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2004, pp. 401–410.

[53] G. Chen et al., "Efficient hierarchical surrogate-assisted differential evolution for high-dimensional expensive optimization," *Inf. Sci.*, vol. 542, pp. 228–246, Jan. 2021, doi: 10.1016/j.ins.2020.06.045.

[54] J. J. Liang et al., "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," *J. Appl. Mech.*, vol. 41, no. 8, pp. 8–31, 2006.

[55] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," Nanyang Tech. Univ. Rep., Singapore, 2010.

[56] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization," Nanyang Tech. Univ. Rep., Singapore, 2017.

[57] B.-C. Wang, H.-X. Li, J.-P. Li, and Y. Wang, "Composite differential evolution for constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 7, pp. 1482–1495, Jul. 2019.

[58] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

Yong Zeng was born in Bijie, China, in 1995. He received the B.S. degree in naval architecture and ocean engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2017, where he is currently pursuing the Ph.D. degree.

His current research interests include surrogate-assisted optimization, and constrained single- and multiobjective optimization.

Yuansheng Cheng was born in Wuhan, China, in 1962. He received the B.S. degree in naval architecture and ocean engineering and the M.S. degree in ship structural mechanics from the Huazhong University of Science and Technology, Wuhan, China, in 1984 and 1987, respectively, and the Ph.D. degree in structural engineering from the University of Hong Kong, Hong Kong, in 2000.

He is currently a Professor with the School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology. His current research interests include structural mechanics, structural impact dynamics, multidisciplinary optimization, and large-scale optimization.

Jun Liu was born in Suizhou, China, in 1981. He received the B.S. and Ph.D. degrees in naval architecture and ocean engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2004 and 2009, respectively.

He is currently a Professor with the School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology. His current research interests include structural optimization, structural impact dynamics, and structural vibration and noise control.