

Surrogate-Assisted Evolutionary Algorithm for Expensive Optimization with Equality and Inequality Constraints

Abstract—The presence of equality and inequality constraints is of significant importance in expensive constrained optimization. This study aims to develop a surrogate-assisted evolutionary algorithm capable of addressing three types of constraints commonly encountered in expensive optimization problems: 1) those with only inequality constraints, 2) those with only equality constraints, and 3) those with both equality and inequality constraints. To achieve this, a radial basis function neural network is employed as a surrogate model to approximate expensive objective functions and individual constraints. Leveraging a consensus measure among all individual constraints, an improved infill sampling criterion is proposed to identify the most promising candidates. In addition, a hybrid local search strategy refines infeasible yet promising solutions via both surrogate-driven evolution and a model-free gradient-based mutation. To balance these search modes cost-effectively, a stagnation strategy enables adaptive switching between exploration and exploitation. Experimental results on 58 test instances demonstrate the efficacy of the proposed approach across all three constraint types under a strict budget of 1000 expensive function evaluations. These findings validate the methodology and show competitive performance relative to state-of-the-art methods.

Index Terms—Expensive constrained optimization, differential evolution, gradient-based mutation, surrogate model.

I. INTRODUCTION

EXPENSIVE constrained optimization problems (ECOPs) are characterized by constraints and a limited computational budget for function evaluations (FEs). A single-objective ECOP, incorporating both inequality and equality constraints, can be formulated as follows:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}), && \mathbf{x} = (x_1, \dots, x_n), \\ &\text{subject to} && g_i(\mathbf{x}) \leq 0, && i = 1, \dots, c_g, \\ & && h_i(\mathbf{x}) = 0, && i = 1, \dots, c_h, \\ & && L_i \leq x_i \leq U_i, && i = 1, \dots, n, \end{aligned} \quad (1)$$

where $\mathbf{x}$ denotes the decision vector; $f(\mathbf{x})$, $g(\mathbf{x})$, and $h(\mathbf{x})$ represent the objective function, inequality constraints, and equality constraints, respectively; L_i and U_i are the lower and upper bounds of the i th variable x_i ; and n , c_g , and c_h indicate the numbers of variables, inequality constraints, and equality constraints, respectively.

A decision vector $\mathbf{x}$ is deemed feasible if it satisfies all constraints, thereby yielding an overall constraint violation of zero; otherwise, $\mathbf{x}$ is infeasible. The overall constraint violation $CV(\mathbf{x})$ is calculated as:

$$CV(\mathbf{x}) = \sum_{i=1}^{c_g} \max\{g_i(\mathbf{x}), 0\} + \sum_{i=1}^{c_h} \max\{|h_i(\mathbf{x})| - \epsilon, 0\}, \quad (2)$$

where $\epsilon = 0.0001$ denotes the tolerance margin for equality constraints [1], ensuring a negligible but finite threshold for

constraint satisfaction. Accordingly, the decision space is partitioned into feasible and infeasible regions. The feasible region specifically encompasses all solutions that meet the constraints.

Within the scope of expensive optimization [2]–[4], the number of FEs is restricted due to the high cost associated with physical experiments and/or computationally intensive simulations. Surrogate-assisted evolutionary algorithms (SAEAs) have emerged as a prominent approach for addressing ECOPs. The basic idea is to build a series of surrogate models to approximate the original expensive objective functions during the evolutionary process [5]. Since the surrogate models are computationally cheaper than the original evaluations for candidate solutions, the cost of evaluations can be significantly reduced. Analogous to model-free evolutionary algorithms (EAs) [1], [6], [7], these surrogate models substitute expensive FEs for evaluating individuals, thereby estimating the most promising candidates through infill sampling criteria to enhance optimization efficiency. Common surrogate models include Gaussian Process (GP) [8], radial basis function networks (RBFNs) [9], [10], support vector machines (SVMs) [11], and gradient boosting classifiers (GBC) [12].

To seek a globally feasible solution within a limited number of expensive FEs, it is imperative that surrogate models effectively guide population evolution. Correspondingly, surrogate model management and utilization in SAEAs are key to addressing the aforementioned issue, which concerns surrogate-driven evolution [13], infill sampling criteria [14], and local search [15]. Surrogate-based evolution, along with its management, is designed to ensure correct evolutionary directions by evaluating individuals through approximate models. Properly managed surrogate-driven evolution enhances convergence accuracy while minimizing computational costs [16]. Following surrogate-driven evolution, a set of candidate individuals is generated. Infill sampling criteria are then applied to pre-screen potential candidate solutions for subsequent expensive evaluations. The criteria usually consider the quality of solutions and their associated uncertainty. For instance, a pre-screened candidate solution that exhibits the lowest overall constraint violation $CV(\mathbf{x})$, when compared to all currently evaluated solutions, is deemed of high quality according to the feasibility rule [17]. Similarly, a pre-screened candidate solution characterized by low prediction confidence represents a potential candidate according to the uncertainty-based rule [18], [19]. Employing expensive FEs for these high-quality solutions with uncertainty can significantly enhance global search capabilities in expensive constrained optimization. Similarly, surrogate-based local search [20] refines given candi-

date solutions using the surrogate outputs, selecting the most promising variant for further expensive evaluation.

Due to the presence of constraints, feasible solutions are preferred over infeasible ones. Thus, handling constraints is pivotal in SAEAs. As alternative FEs, it is imperative that constraints are considered within the surrogate model itself. Commonly, approximations can suffer from overfitting or underfitting, significantly reducing the accuracy of surrogate models for constraints and potentially misleading the evolution in handling them. When multiple constraints are involved, the feasible region becomes exceedingly narrow, making it difficult to extract information for feasible solutions from the training data [19]. Unlike approximations for a single objective function, the goal here is to build efficient training models that account for all constraints, presenting a more challenging problem. To mitigate the difficulties posed by constraints, special surrogate model-building and data selection strategies have been developed [21]. For example, to explore different regions of interest for objectives and constraints, Liu *et al.* [22] and Rahi *et al.* [23] constructed a series of surrogate models for the objective function and each constraint. In [19], [24], data representing the overall constraint violation has been used to develop an alternative surrogate model for all constraints together, simplifying the constraint approximation. To balance uncertainty and accuracy, Wang *et al.* [20] developed an effective combination strategy utilizing both cheap and expensive surrogates. At each generation, all expensively evaluated decision vectors are used to construct a cheap surrogate. Meanwhile, for each expensive surrogate, K nearest neighbors to the targeted decision vector are selected as training data.

During the evolutionary process, surrogate models need to be updated by adding new candidate solutions (samples) selected according to the infill sampling criterion. As a model management strategy, various infill sampling criteria have been proposed for ECOPs. Yang *et al.* [25] proposed generating a set of offspring via the multiple offspring generators and select the most promising individual based on improved feasibility rules, which consider not only the overall constraint violation but also the number of constraint violations. Therefore, the selected sample can significantly reduce misjudgment regarding the superiority of infeasible solutions due to the inconsistency of the magnitudes for different constraints. In [10], candidate solutions are pre-screened based on potential and uncertainty. The distances between the predicted optimal solution and all expensively evaluated individuals are calculated to automatically decide whether to choose the potentially best solution or the one with high uncertainty for expensive evaluation. In this way, candidate solutions that are similar to the current best solution can be avoided. Wei *et al.* [19] developed a feasible exploration strategy to pre-screen candidate solutions, which is efficient for problems with complicated feasible areas. The feasible exploration uses DE/current-to-feasible/1 as the search optimizer to generate trial samples. After few generations, the feasibility rule combined with the penalty-based technique is used to select the predicted best candidate.

To date, most research on SAEAs has focused on evolutionary strategies for ECOPs with inequality constraints, whereas comparatively few studies have addressed surrogate-model

management, infill sampling criteria, and surrogate-driven evolution tailored to ECOPs with equality constraints. According to (1), ECOPs can be categorized into three types: 1) those with only inequality constraints, 2) those with only equality constraints, and 3) those with both equality and inequality constraints. For the latter two types, equality constraints typically confine the feasible set to a thin manifold and confine tight feasibility tolerances (e.g., precision of at least 0.0001), substantially increasing search difficulty. Consequently, SAEAs that perform well on inequality-only ECOPs often degrade when equality constraints are present. Given the prevalence of equality-constrained ECOPs in practice, there is a clear need to close this gap under limited budgets of expensive FEs. Moreover, strategies for surrogate-model management, infill sampling, and model-free local optimization—though effective and successful on inequality-only ECOPs—may not be suitable for the latter two ECOP types considered here. Motivated by these observations, this study develops tailored mechanisms for surrogate model management, a consensus-aware infill sampling criterion, and a cost-effective reuse strategy for gradient-based mutation as local search to solve ECOPs spanning all three constraint types. Our main contributions are as follows:

- We introduce a stagnation strategy that coordinates global and local surrogate-driven evolution with a model-free optimizer. Surrogate-driven evolution provides cost-effective exploration of feasibility with few expensive FEs, whereas the model-free local optimizer offers accurate exploitation near tight equality manifolds. The stagnation trigger enables a controlled, cost-aware handoff between these modes.
- We propose an infill criterion that first seeks consensus among individual constraint improvements before considering candidates based on objective value. Coupled with the stagnation policy, this criterion improves surrogate utilization and reduces unnecessary invocations of the model-free local optimizer.
- Building on classical gradient-based mutation [30], we design a reuse strategy for gradient-based local search, namely gradient-reused mutation (GRM), targeted at difficult constraints. Gradient information that has successfully improved an infeasible solution is reapplied until no further progress is observed, enabling substantial feasibility gains under tight FE budgets.
- Two test suites covering all three ECOP types are used to assess the method under highly restricted FE budgets. Against six state-of-the-art SAEAs, the proposed approach delivers substantial improvements across diverse constraint configurations, indicating a promising direction for tackling more complex, computationally demanding ECOPs.

The remainder of this paper is organized as follows. Section II provides a brief overview of the preliminaries. The proposed SAEA is thoroughly discussed in Section III. Section IV presents empirical experiments of the proposed algorithm, providing a detailed analysis. Finally, Section V concludes the paper and outlines future work.

II. PRELIMINARIES

In this section, we briefly introduce the RBFN surrogate model, DE, and gradient-based mutation.

A. Feasibility Rule

For constrained optimization, whether it is expensive or not, the focus is primarily on feasible solutions, as infeasible solutions inherently lack optimality. Therefore, handling constraints is a critical aspect of constrained optimization within EAs. Among various constraint handling techniques, the feasibility rule [17] is the most popular. It not only demonstrates powerful capabilities but also operates without parameter control. Specifically, three criteria are used to compare two candidate solutions, as follows:

- 1) Any feasible candidate solution is preferred over any infeasible one.
- 2) If both two candidate solutions are infeasible, the one with a smaller overall constraint violation is preferable.
- 3) If both two candidate solutions are feasible, the one with a better objective function value is preferable.

The first two criteria drive the evolutionary population toward feasible regions, while the third assists in exploiting high-quality feasible solutions. Previous studies show that the feasibility rule can handle constraints at a significantly fast speed. However, since the first two criteria employ a greedy strategy, there is a risk of becoming stuck in local optima. To mitigate this issue, Wang *et al.* [31] propose an efficient replacement mechanism to replace a certain number of infeasible solutions that have worse objective function values, despite having smaller overall constraint violations.

On one hand, the feasibility rule can quickly guide the evolution toward the feasible region. This is particularly important in expensive constrained optimization, especially when equality constraints are involved, as the feasible region becomes significantly narrow, and fast convergence is desirable due to the limited computational budget. On the other hand, the approximation of constraints by surrogates can inevitably cause distortion or loss of quality. The challenge of employing the feasibility rule in a robust manner to avoid performance degradation while handling ECOPs with all types of constraints remains an area requiring further investigation.

B. Radial Basis Function Network

The RBFN was first formulated by Broomhead and Lowe [32] in 1988. In this model, radial basis functions serve as activation functions, and the output is a linear combination of such functions applied to the inputs and neuron parameters. In this study, the RBFN with a cubic form is adopted as a surrogate, and the approximation output y can be expressed as follows:

$$y = \mathbf{w}^T \boldsymbol{\varphi} = \sum_{i=1}^N \omega_i \phi(\|\mathbf{x} - \mathbf{x}_i\|^3) \quad (3)$$

where, N is the cardinality of the training set, $\mathbf{w}$ is an $N \times 1$ weight vector including $\omega_i, i = 1, 2, \dots, N$, $\boldsymbol{\varphi}$ is another $N \times 1$ vector containing the values calculated by the basis

function of cubic form $\phi(\cdot)$, and the norm $\|\cdot\|$ is taken to be the Euclidean distance. The cubic form is commonly used for $\phi(\cdot)$ in RBFNs and their variants [10], [20], [25] due to its effectiveness and simplicity, and this study adopts this form as well. The unknown $\mathbf{w}$ can be calculated as follows:

$$\mathbf{w} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{F} \quad (4)$$

where $\mathbf{F} = (f(\mathbf{x}_1), (\mathbf{x}_2), \dots, (\mathbf{x}_N))^T$ and $\boldsymbol{\Phi}$ is the Gram matrix defined as

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi(\mathbf{x}_1 - \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_1 - \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N - \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_N - \mathbf{x}_N) \end{bmatrix} \quad (5)$$

Due to its strong performance, the RBFN has become a popular choice for approximating black-box functions. Furthermore, its characteristic of parameter insensitivity during neural network training has made it especially attractive to researchers and users. In the field of expensive optimization, the RBFN has been one of the most widely used surrogates for approximating objective functions and constraints. For example, Regis [33] utilized all collected data to train RBFN surrogates for direct approximation of both objective functions and constraints. In contrast, Li and Zhang [34] adopted multiple local RBFN surrogates to approximate a single constraint.

C. Differential Evolution

As the search engine, differential evolution (DE) [35] and its variants employ mutation and crossover operators to generate the trial vectors of offspring.

Suppose that the population P consists of NP decision vectors of individuals, defined as $P = \{\mathbf{x}_i, \dots, \mathbf{x}_{NP}\}$. First, the mutation operator produces the mutant vector $\mathbf{v}_i$ for the targeted individual $\mathbf{x}_i$, where $i \in [1, NP]$. In this study, five of the most popular mutation operators used in constrained optimization [10], [25], [36] are adopted:

- 1) DE/best/2

$$\mathbf{v}_i = \mathbf{x}_{best} + F_{sc}(\mathbf{x}_{r1} - \mathbf{x}_{r2}) + F_{sc}(\mathbf{x}_{r3} - \mathbf{x}_{r4}), \quad (6)$$

- 2) DE/rand/2

$$\mathbf{v}_i = \mathbf{x}_{r1} + F_{sc}(\mathbf{x}_{r2} - \mathbf{x}_{r3}) + F_{sc}(\mathbf{x}_{r4} - \mathbf{x}_{r5}), \quad (7)$$

- 3) DE/rand-to-best/1

$$\mathbf{v}_i = \mathbf{x}_{r1} + F_{sc}(\mathbf{x}_{best} - \mathbf{x}_{r1}) + F_{sc}(\mathbf{x}_{r2} - \mathbf{x}_{r3}), \quad (8)$$

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

$$\mathbf{v}_i = \mathbf{x}_i + F_{sc}(\mathbf{x}_{best} - \mathbf{x}_i) + F_{sc}(\mathbf{x}_{r1} - \mathbf{x}_{r2}), \quad (9)$$

- 5) DE/current-to-rand/1

$$\mathbf{v}_i = \mathbf{x}_i + F_{sc}(\mathbf{x}_{r1} - \mathbf{x}_i) + F_{sc}(\mathbf{x}_{r2} - \mathbf{x}_{r3}), \quad (10)$$

where $r1, r2, r3, r4$ and $r5$ are distinct integers randomly selected from the range $\{1, \dots, i-1, i+1, \dots, NP\}$, $\mathbf{x}_{best}$ is the decision vector of the best individual found so far, and F_{sc} is the scaling factor.

After mutation, the crossover operator is applied to produce the trial vector $\mathbf{u}_i$ based on $\mathbf{v}_i$ and $\mathbf{x}_i$. The commonly used binomial crossover is defined as follows:

$$\mathbf{u}_{i,j} = \begin{cases} \mathbf{v}_{i,j}, & \text{if } \text{rand}_j \leq CR \text{ or } j = j_{rand}, \\ \mathbf{x}_{i,j}, & \text{otherwise.} \end{cases} \quad j = 1, \dots, n, \quad (11)$$

where j_{rand} is an integer from $\{1, \dots, n\}$, rand_j is a decimal from $[0, 1]$, and CR is the crossover rate.

D. Gradient-based Mutation

Gradient descent is a premier technique for numerical optimization. It locates the optimum by iteratively moving against the gradient of the targeted objective function at a given point. Recently, gradient descent has been developed as a repair method or local search [22] for model-free constrained optimization to rapidly seek the feasible region. The basic idea is derived from the first-order Taylor expansion:

$$C(\mathbf{x}') \approx C(\mathbf{x}) + \nabla_{\mathbf{x}}G(\mathbf{x}' - \mathbf{x}), \quad (12)$$

where $C(\mathbf{x})$ is the vector for all equality and inequality constraint violations. $\nabla_{\mathbf{x}}G$ is the Jacobian matrix for $C(\mathbf{x})$, and $\mathbf{x}$ and $\mathbf{x}'$ are the initial and refined decision vectors, respectively.

The term $\nabla_{\mathbf{x}}G$ indicates the direction of the fastest increase of the constraints. Conversely, its opposite, $-\nabla_{\mathbf{x}}G$, points to the steepest descent direction along the constraints. After gradient-based mutation, violations are expected to be repaired. To this end, the refined decision vector $\mathbf{x}'$ is fine-tuned to be feasible as follows:

$$\begin{cases} g_i(\mathbf{x}') \leq 0, & i = 1, \dots, c_g, \\ h_i(\mathbf{x}') = 0, & i = 1, \dots, c_h. \end{cases} \quad (13)$$

According to (1) and (2), the constraint violation is always considered to be positive, and in this case, $C(\mathbf{x})$ is effectively a zero matrix. Therefore, under these conditions, (12) can be simplified

$$\mathbf{x}' = -\nabla_{\mathbf{x}}^{-1}GC(\mathbf{x}) + \mathbf{x} \quad (14)$$

Note that if $\nabla_{\mathbf{x}}G$ is not invertible, the Moore-Penrose inverse or pseudo-inverse is employed to compute $\nabla_{\mathbf{x}}^{-1}G$.

The gradient-based mutation and its variants have been successfully applied to model-free constrained optimization, achieving outstanding performance in many benchmark competitions and real-world applications. Despite their efficacy, there are two key considerations when applying gradient-based mutation. First, gradient-based mutation is only conducted on infeasible solutions with constraint violations greater than zero; for the constraints that have been satisfied, their gradients are not utilized in (14). This approach ensures that the elements of $\nabla_{\mathbf{x}}G$ are not zero, thus largely avoiding the vanishing gradient problem. Second, due to the black-box nature of the optimization process, n FEs are often required to numerically calculate the gradient matrix by finite difference method; therefore, an additional scheduling scheme is necessary to avoid wasting FEs.

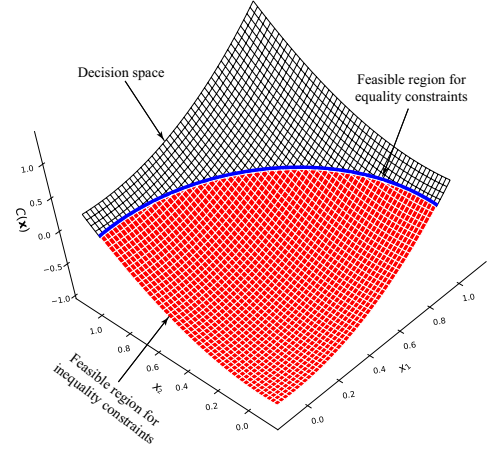


Fig. 1. A simple illustration of the increased difficulty in handling equality constraints. The large red region representing the feasible region for inequality constraints is significantly reduced to the narrow blue curve when these constraints are transformed into equality constraints.

III. PROPOSED APPROACH

In this section, we first elaborate on the motivation for this study and then detail the proposed SAEA for ECOPs with different types of constraints.

A. Motivations

Many real-world optimization problems are constrained not only by inequalities but also by equalities. Furthermore, their FEs often depend on costly physical experiments or time-consuming simulations. To date, state-of-the-art SAEAs for expensive constrained optimization have been successfully developed to handle the first type of constraints, specifically ECOPs with inequalities. However, addressing ECOPs with the other two types of constraints is also crucial. Given specific application scenarios, the consideration of equalities in expensive constrained optimization remains necessary. Therefore, this study focuses on developing an SAEA to meet this need.

The search process for ECOPs with equality constraints significantly differs from that for inequalities, presenting additional challenges for expensive constrained optimization. To illustrate this difficulty, we utilize test function G03 from the CEC2006 benchmark set [37]:

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) = -(\sqrt{n})^n \prod_{i=1}^n x_i, \\ & \text{subject to} && h_1(\mathbf{x}) = \sum_{i=1}^n x_i^2 - 1 = 0, \end{aligned} \quad (15)$$

where $n = 10$ and $0 \leq x_i \leq 1, i = 1, \dots, 10$. For comparison, we reformulate the above equality constraint into an inequality constraint:

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) = -(\sqrt{n})^n \prod_{i=1}^n x_i, \\ & \text{subject to} && g_1(\mathbf{x}) = \sum_{i=1}^n x_i^2 - 1 \leq 0. \end{aligned} \quad (16)$$

The decision space and feasible regions for (15) and (16) are depicted with the first two variable x_1 and x_2 in Fig. 1. Although we maintain the decision space and the mathematical

expressions for the objective function and constraints, changing only the inequality sign “ $\leq$ ” to the equality sign “ $=$ ”, the feasible region of the equality constraint is substantially narrower than that of the inequality constraint, as evidenced by the red area and the narrow blue curve. Moreover, compared to the entire decision space, the slim feasible region of inequalities poses a greater challenge for identification within a large decision space via surrogate-driven evolution. Based on these observations, locating the feasible solution with the minimum objective function value becomes difficult. Therefore, when considering all three types of constraints in the design of SAEAs, there is a need to further develop existing model management, surrogate utilization, and infill sampling criteria to address ECOPs with equalities.

Inspired by the aforementioned motivations, we propose a surrogate-assisted DE with stagnation activated GRM (SaDE-SA-GRM) to solve ECOPs with all three types of constraints. The schematic diagram of SaDE-SA-GRM is illustrated in Fig. 2. In this approach, all collected data are used to construct global RBFN surrogate models for approximating the objective function as well as equality and/or inequality constraints. Once the surrogate model is adequately trained, it is employed to guide the population over a specified number of generations. Consistent with most SAEAs, the primary goal of the proposed global RBFN surrogate-driven evolution is to initially explore the feasible region and subsequently identify the promising solution via our developed consensus-aware infill sampling criterion. As previously noted, equality constraints typically result in extremely narrow feasible regions that are challenging to explore effectively. To address this issue, we integrate the local RBFN surrogate-driven evolution with the model-free gradient-based optimizer, i.e., GRM, to improve exploitation capabilities. To overcome the high expensive FE cost typically associated with numerical gradient computation [30], two specialized strategies for surrogate model management are introduced. The first strategy activates GRM upon detecting stagnation in surrogate-driven evolution, thereby enhancing exploration efficiency. The second strategy employs a reuse strategy, allowing the model-free optimization method to be repeatedly applied in a cost-effective manner, further reducing computational overhead.

In the following subsections, we provide a detailed discussion of RBFN surrogate-driven evolution, the developed infill sampling criteria, GRM, and their implementation management, including the proposed stagnation strategy.

B. RBFN Surrogate-Driven Evolution

When considering equality constraints, the fitness landscape becomes more complex, necessitating a highly accurate and generalizable approach for approximation. Therefore, the RBFN, known for its high approximation accuracy and excellent generalization ability [33], is chosen as a surrogate in our proposed method. On the other hand, population-based EAs progress incrementally, with significant improvements accumulating over generations. For expensive unconstrained optimization, generation-based evolution control ensures the correct convergence of SAEAs [16]. However, due to the

presence of feasible regions, the convergence of SAEAs for expensive constrained optimization is more complex. Two different types of convergence are asynchronously required for ECOPs: correct convergence toward a feasible region within the decision space and fast convergence toward the best feasible solution within the feasible region. To achieve this, we build global and local surrogate models to drive the generation-based global and local evolution. As the procedures for building global and local surrogates are the same, with the only difference being the input information, we first outline a general framework in **Algorithm 1**, implementation details are given sequentially.

Algorithm 1: Surrogate-Driven Evolution

Input:

- TS : the training set to build a surrogate model;
- P : the set of NP individuals for surrogate-driven evolution;

Use TS to train the RBFN for the objective function and constraints;

while Tm generations are not achieved **do**

for x in P **do**

 Generate an offspring u using a certain DE operator;

 Evaluate the offspring u by the well-trained RBFN;

 Update x by u according to the feasibility rule.

Output:

The individuals in P .

For global surrogates, the archive containing all expensively evaluated solutions is used as the training set TS to train the RBFN. The current population, consisting of NP individuals, serves as the initial population for the global surrogate-driven evolution. The search space is defined as the problem decision space, $L_i \leq x_i \leq U_i, i = 1, \dots, n$. For local surrogates, the best NP individuals are selected from the archive to form the TS and P , defining their occupied space—the range between the maximum and minimum values of each decision variable—as the search space. Subsequently, the for loop operates analogously to most model-free EAs. The outputs of well-trained surrogates substitute expensive FEs to drive the population-based evolution within Tm generations. Specifically, for each offspring u , a DE operator is randomly selected from the five mutation operators introduced in Subsection II-C, while F_{sc} and CR are generated as $0.5rand + 0.5$ as referenced in [25], [38]. The values of the objective function, constraint violations, and overall constraint violation are predicted by the surrogate. To update x by u , the standard feasibility rule introduced in Section II-A is applied. After the while loop, the evolved population P is output for further infill sampling.

C. Developed Infill Sampling Criterion

By global RBFN surrogate-driven evolution, NP individuals are obtained. The selection of the most promising candidate solution for updates is crucial for rapidly exploring the feasible

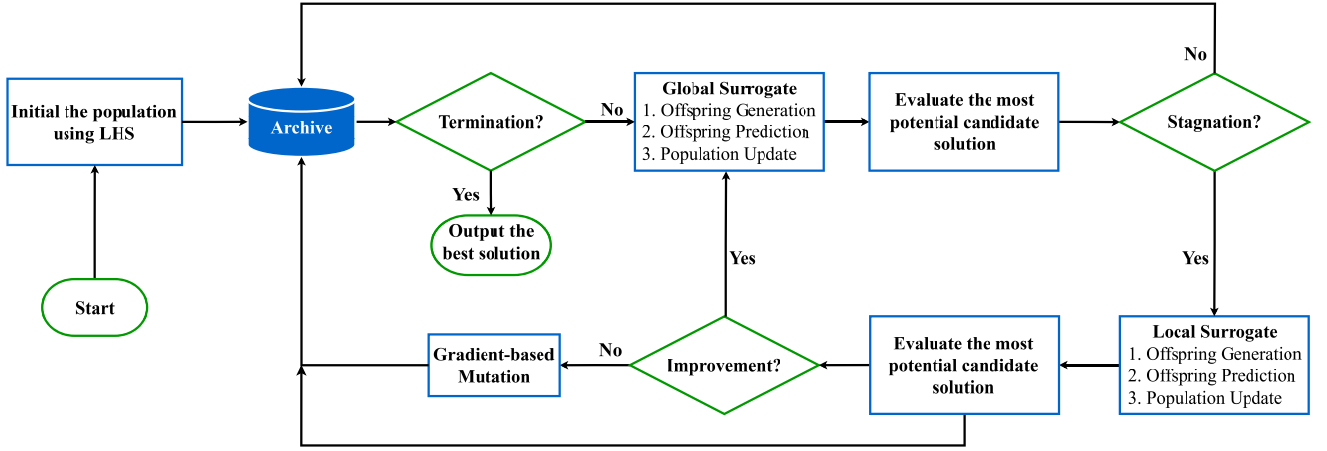


Fig. 2. The schematic diagram of SaDE-SA-GRM.

region. Additionally, SAEAs for ECOPs typically perform fewer FEs, limiting the ability to adjust the evolutionary direction by the trial-and-error method. Therefore, the infill sampling criterion plays a vital role in preventing premature convergence, especially within the narrow feasible region caused by equality constraints.

So far, numerous studies [28], [29], [31], [39] have confirmed that the objective function can mitigate the search bias introduced by constraint handling. As highlighted in [28], the correlation between objective function optimization and constraint satisfaction has been shown to direct evolutionary search toward different promising regions. Moreover, due to limited opportunities for trial and error, a consensus between improvements in all constraints can significantly reduce the bias caused by substantial improvements in one or two specific constraints. Inspired by these insights, we have developed a new infill sampling criterion to identify potentially good candidate solutions. As detailed in **Algorithm 2**, when comparing the i th pairwise individuals $\mathbf{q}_i$ and $\mathbf{p}_i$, if $\mathbf{q}_i$ is no worse than $\mathbf{p}_i$ across all individual constraints, $\mathbf{q}_i$ is stored in S . If such a consensus is not achieved by any pairwise comparison, and thus S remains empty, the overall constraint violation $CV(\mathbf{x})$ is used to compare $\mathbf{q}_i$ and $\mathbf{p}_i$ in the subsequent for loop. After employing these two comparative strategies, if S is not empty, the individual with the minimum objective function value in S is selected for further expensive evaluation; otherwise, a stagnant flag is then output.

D. GRM

Recent research [22], [40] has demonstrated the overwhelming effectiveness of gradient-based mutation in handling constraints. This approach repairs infeasible solutions by gradient information, enabling rapid movement towards the feasible region. Whether the optimization is expensive or not, constraints share similar characteristics; thus, gradient-based mutation is also capable of managing constraints in ECOPs. However, a notable drawback of gradient-based mutation is its high computational cost, as it requires n FEs to obtain the gradient matrix, which is particularly burdensome in expensive optimization scenarios. Nonetheless, as indicated

Algorithm 2: Developed Infill Sampling Criterion

Input:

- the original population: $P = \{\mathbf{p}_1, \dots, \mathbf{p}_{NP}\}$;
- the newly evolved population: $Q = \{\mathbf{q}_1, \dots, \mathbf{q}_{NP}\}$;
- the associated parameters: c_g and c_h ;

Set $S = \{\}$;

for $i = 1$ **to** NP **do**

if $g_j(\mathbf{q}_i) \leq g_j(\mathbf{p}_i), j = 1, \dots, c_g$ **and**
 $h_t(\mathbf{q}_i) \leq h_t(\mathbf{p}_i), t = 1, \dots, c_h$ **then**
 $S = S \cup \{\mathbf{q}_i\}$;

if S is empty **then**

for $i = 1$ **to** NP **do**

if $CV(\mathbf{q}_i) \leq CV(\mathbf{p}_i)$ **then** $S = S \cup \{\mathbf{q}_i\}$;

if S is not empty **then**

Perform an expensive evaluation of the individual with the minimum objective value in S (denoted $\mathbf{q}$);

Identify the individual in P with the maximum overall constraint violation (denoted $\mathbf{p}$);

if $\mathbf{q}$ is preferred to $\mathbf{p}$ according to the feasibility rule **then**

Replace $\mathbf{p}$ with $\mathbf{q}$ and output P ;

else Output stagnation;

else Output stagnation;

in [13], model-free local searches can enhance the potential of surrogate models and mitigate their adverse effects. Therefore, it is worthwhile to explore how to economically employ gradient-based mutation to achieve fast convergence in SAEAs. To address this issue, we propose two strategies: one restricts the implementation of gradient-based mutation to a significantly low level through the stagnation of evolution, and the other maximizes convergence speed through a reuse strategy. Since stagnation is applied globally in SaDE-SA-GRM, its implementation is detailed in the subsection on the overall approach. This subsection describes the GRM.

As outlined in Section III-B, SaDE-SA-GRM encompasses

both global and local surrogate-driven evolution. If neither search strategy finds any improvement under certain conditions, the evolution is considered stagnant, and then gradient-based mutation is activated as an alternative search strategy to repair infeasible solutions. The adaptive approach to GRM is presented in **Algorithm 3**. Given an infeasible solution $\mathbf{x}$, its gradient matrix $\nabla_{\mathbf{x}}G$ is numerically calculated using n expensive FEs. Subsequently, $\nabla_{\mathbf{x}}G$ is repeatedly used to update $\mathbf{x}$ and its successor until no further improvements are obtained (see the if-else condition). The rationale for reusing $\nabla_{\mathbf{x}}G$ includes:

- It is impractical to frequently calculate the gradient matrix due to the consumption of n expensive FEs from the total computational budget.
- The primary objective of gradient-based mutation is to convert violations into feasible solutions. The efficiency of gradient-based mutation derived from Eqs. (12) to (14) increases as the infeasible solution approaches the feasible region, thus justifying the reuse of the gradient matrix to maximize improvement.
- Considering the vanishing gradient problem, the closer the solution is to the feasible region, the more likely this issue occurs. Therefore, compared to $\mathbf{x}$, $\mathbf{x}'$ is more prone to incur vanishing gradients if the gradient matrix at $\mathbf{x}'$ is utilized.

An example (Appendix A in the supplementary material) is given to mathematically describe the employment of GRM.

Algorithm 3: GRM

Input:

- the input infeasible solution: $\mathbf{x}$;
- the associated parameter: $nFEs$;

Numerically calculate the gradient matrix at $\mathbf{x}$ as $\nabla_{\mathbf{x}}G$;

Archive the n intermediate solutions;

Set $nFEs = nFEs + n$;

while $\mathbf{x}$ is infeasible **do**

 Use $\nabla_{\mathbf{x}}G$, $\mathbf{x}$, and $C(\mathbf{x})$ to calculate $\mathbf{x}'$ via (14);

 Evaluate $\mathbf{x}'$;

 Archive $\mathbf{x}'$;

 Set $nFEs = nFEs + 1$;

if $CV(\mathbf{x}') \leq CV(\mathbf{x})$ **then**

 Set $\mathbf{x} = \mathbf{x}'$, $C(\mathbf{x}) = C(\mathbf{x}')$, and $f(\mathbf{x}) = f(\mathbf{x}')$;

else break;

Output: $\mathbf{x}$;

E. Implementation with Stagnation Strategy

The overall approach is presented in **Algorithm 4**. SaDE-SA-GRM is generally executed in five main steps as follows:

Step 1 Initialization: NP decision vectors are generated using Latin hypercube sampling (LHS) [41] in the decision space. These NP decision vectors are then evaluated using expensive FEs and collected into an archive A . Meanwhile, their copies constitute the initial population

Algorithm 4: SaDE-SA-GRM

Initialization;

Set stagnation count $stg = 0$;

while $nFE < MaxFEs$ **do**

 Execute **Algorithm 1** for global surrogate-driven evolution;

 Execute **Algorithm 2** for infill sampling;

if stagnation is output **then**

 Set $stg = stg + 1$;

else Set $stg = 0$;

if $stg = s_{num}$ **then**

 Execute **Algorithm 1** for local surrogate-driven evolution;

 Perform an expensive evaluation to the best individual and observe the improvement;

if no improvement **and** no feasible solution found **then**

 Execute **Algorithm 3** for GRM;

 Set $stg = 0$;

Output: best solution;

P . It is worth noting that, throughout the evolutionary process, each expensive evaluation is recorded and its corresponding results are archived in A .

Step 2 Global Surrogate-Driven Evolution: RBFN models are built using all the expensively evaluated data in A . Since all archived information is used to train this surrogate, the approximation to the whole fitness landscape of the objective function and all individual constraints is expected to be established. Therefore, RBFN surrogate models are used to globally replace the original FEs to drive the population P over T_m generations.

Step 3 Infill Sampling: Following the global RBFN surrogate-driven evolution, the most promising candidate is selected for expensive evaluation. The proposed infill sampling criterion is used to (i) detect consensus among improvements across all individual constraints and (ii) account for objective optimization. If no candidate satisfies above criterion, or if the expensively evaluated candidate fails to replace the worst individual in the current population, the stagnation flag is output and the stagnation counter stg is incremented; otherwise reset stg to zero. When stagnation persists for s_{num} consecutive iterations, proceed to **Step 4**; otherwise, return to **Step 2**.

Step 4 Local Surrogate-Driven Evolution: RBFN models are constructed to exploit the best solution $\mathbf{x}_{bs}$ found so far. First, $\mathbf{x}_{bs}$ and its $NP - 1$ nearest neighbors from A serve as both the training set and the initial population for local surrogate-driven evolution. Next, the RBFN surrogates drive these NP individuals for T_m generations. The original feasibility rule is then applied to select the best candidate for expensive evaluation. If no improvement is achieved by the local surrogate-driven evolution and no feasible solution has yet been found, proceed to **Step 5**; otherwise, reset stg to zero and return to **Step 2**.

Step 5 GRM: As a model-free local search, gradient-based mutation is invoked only when no feasible solution has

been found and is applied to the best infeasible solution. Upon activation, our proposed GRM described in Section II-D is employed to further refine this solution. Regardless of whether an improvement is achieved, stg is reset to zero, and the process returns to **Step 2**.

As mentioned in Section III-D, stagnation aims to economically employ gradient-based mutation. Here, we introduce s_{num} to control the frequency of using local surrogate-driven evolution and GRM. The number of consecutive stagnation events caused by global surrogate-driven evolution is counted throughout the evolutionary process, as shown by stg in **Algorithm 4**. For local RBFN-driven evolution, the goal is to refine the infeasible solution using local information, rather than to explore different promising and interesting regions. Thus, the original feasibility rule is adopted. As a refinement step, local RBFN-driven evolution is repeated until no further improvements can be achieved. The variable $nFEs$ tracks the number of expensive FEs performed to date, including both candidate-solution evaluations and numerical gradient computations via the finite-difference method. The maximum allowable FEs, denoted $MaxFEs$, are used as the termination criterion for SaDE-SA-GRM.

IV. EMPIRICAL STUDY

In this section, extensive experiments are conducted to demonstrate the capabilities of the proposed SaDE-SA-GRM. First, we briefly introduce the experimental settings for our empirical studies, including test instances, parameter settings, and the running environment. Second, we investigate the sensitivity of parameter settings for both surrogate-driven evolution and stagnant evolution. Third, we examine the influences of the developed infill sampling criterion and the GRM. Finally, we compare the performance of SaDE-SA-GRM with five state-of-the-art SAEAs to verify the advantages of handling ECOPs. Additionally, the time complexity is empirically investigated in this section.

A. Experiment Settings

1) *Test Instances*: Fifty-eight test instances from the two well-known test sets, CEC2006 [37] and CEC2010 [42] are adopted. In this study, the objective functions and constraints of these 58 test instances are assumed to be expensive. Thus, the $MaxFEs$ is set to 1000 for each test instance. Characteristics of these test instances, such as dimensions and feasibility regions, are presented in Table S.I and Table S.II (supplementary material). It is worth mentioning that the feasibility region ρ is defined as the ratio between the feasible region and the decision space; a value of 0.0000% indicates that ρ is smaller than 0.0001%.

2) *Parameter Settings*: Regarding SaDE-SA-GRM, the population size P_S is set to 100. The termination parameter T_m for surrogate-driven evolution is set to 200. The maximum number of consecutive stagnations, denoted as s_{num} , is set to 5. For the empirical study, 20 independent runs are conducted on each test instance. To ensure consistent results for both the CEC2006 and CEC2010 test instances, the averages of the function values (AFV) and overall constraint violations

(AOCV) obtained by the corresponding approach are presented. Additionally, the success ratio (SR), which indicates the proportion of successful runs over the total number of runs, is calculated for comparison. The Wilcoxon signed-rank test and the Friedman aligned test with the Hommel post-hoc method [43] are performed at a significance level of 0.05 for non-parametric analyses.

3) *Running Environment*: SaDE-SA-GRM and its variants were implemented using Python 3.8.10. All empirical experiments were executed on a workstation equipped with an AMD Ryzen 9 5900X 12-Core Processor and 64.0 GB of memory, running on the Windows 10 operating system.

B. Sensitivity of Parameter Settings

There are two specially designed parameters, namely T_m and s_{num} , in SaDE-SA-GRM. Different settings of T_m and s_{num} may significantly affect the evolutionary ability. Thus, we adopt a series of values for $T_m \in \{1, 20, 50, 100, 200, 300\}$ and $s_{num} \in \{1, 5, 10\}$ to analyze the sensitivity of these two parameters on the CEC2006 test set.

1) *Sensitivity of T_m* : Table S.III (supplementary material) provides the empirical results of SR and AFV obtained by SaDE-SA-GRM with different T_m settings on 22 test instances. The best SR and AFV obtained on each test instance are highlighted in **bold**. It can be observed that SaDE-SA-GRM with $T_m = 300$ achieves the best overall performance across the 22 test instances, while SaDE-SA-GRM with $T_m = 1$ ranks the worst. As the termination criterion for surrogate-driven evolution, T_m determines the search depth driven by the corresponding RBFN surrogate. Analogous to model-free EAs, a large value of T_m results in prolonged population evolution, allowing for deeper exploration of the surrogate. When T_m is set to a smaller value, such as 1, 20, 50, or 100, the explorations achieved by both global and local surrogate-driven evolution are relatively shallow. For example, 12 out of 22 test functions cannot be solved by SaDE-SA-GRM with $T_m = 1$ in 20 consecutive independent runs. On the other hand, although the performance of SaDE-SA-GRM may improve with a value larger than 300 for T_m , the associated time complexity becomes unacceptable for expensive optimization (see the analysis in Subsection IV-F). Therefore, considering optimal performance and time consumption simultaneously, we set $T_m = 200$ for all test instances in this study.

2) *Sensitivity of s_{num}* : Table S.IV (supplementary material) presents the comparison results of SaDE-SA-GRM with $s_{num} \in \{1, 5, 10\}$ on 22 test instances. The best result for each test instance is highlighted in **bold**. It can be observed that SaDE-SA-GRM with $s_{num} = 5$ achieves the best performance on 15 out of 22 test instances. In contrast, SaDE-SA-GRM with $s_{num} = 1$ and $s_{num} = 10$ performs best on 11 and 9 test instances, respectively. The parameter s_{num} controls how often local surrogate-driven evolution and the proposed GRM are invoked; conversely, it also determines how many additional attempts are allowed to global surrogate-driven evolution. With $s_{num} = 1$, GRM is triggered immediately in the early and middle stages whenever the global surrogate-driven evolution fails to produce a promising candidate. In this

case, a substantial number of expensive FEs are consumed to employ local surrogate-driven evolution and compute the gradient vector and Jacobian matrix for GRM. Accordingly, frequent local-search and GRM invocations are not cost-effective when individuals are far from the feasible region, and moreover, the induced rapid convergence increases the risk of premature convergence and can limit global exploration. With $s_{num} = 10$, SaDE-SA-GRM adopts a higher tolerance for repeating global surrogate-driven evolution, making exploration the dominant mode for seeking feasibility. Local surrogate-driven evolution and GRM are then deferred to later stages. Under this setting, the algorithm may lack sufficient budget to fully exploit the feasible region and approach the global optimum. Based on empirical results, we set $s_{num} = 5$ in SaDE-SA-GRM to balance exploration and exploitation.

C. Influence of the Proposed Infill Sampling Criterion

The proposed infill sampling criterion selects a promising candidate in two steps. First, pairwise comparisons are performed via the two for loops in **Algorithm 2**. Preference is given to achieving consensus across all individual constraints in the first loop; if no such consensus is found, the overall constraint violation is used in the second loop. Second, among the remaining candidates, the one with the minimum objective function value is selected. To assess the impact of these two steps, we consider two variants:

- 1) SaDE-SA-GRM-CVOB: Omits the consensus-based pairwise comparison in the first loop. Instead, the overall constraint violation is used directly in each pairwise comparison; subsequently, the candidate with the minimum objective function value is selected.
- 2) SaDE-SA-GRM-CssCV: Retains both for loops but applies the overall constraint violation to select the promising candidate—that is, the solution with the minimum overall constraint violation; in case of ties, the one with the smaller objective function value is chosen.

Pseudo-code for SaDE-SA-GRM-CVOB and SaDE-SA-GRM-CssCV is provided in **Algorithm S.1** and **Algorithm S.2** (supplementary material), respectively. Table S.V (supplementary material) reports the empirical SR and AFV obtained by SaDE-SA-GRM-CVOB, SaDE-SA-GRM-CssCV, and SaDE-SA-GRM on the CEC2006 test set, with the best results highlighted in **bold**. Nonparametric analyses are also included. Symbols “+”, “−”, and “ $\approx$ ” indicate that SaDE-SA-GRM-CVOB or SaDE-SA-GRM-CssCV is significantly better than, worse than, or statistically indistinguishable from SaDE-SA-GRM according to the Wilcoxon rank-sum test at a significance level of 0.05.

Compared with SaDE-SA-GRM-CVOB and SaDE-SA-GRM-CssCV, SaDE-SA-GRM attains superior performance on a number of test instances across all three constraint settings. As shown in Table S.V, SaDE-SA-GRM significantly outperforms SaDE-SA-GRM-CVOB and SaDE-SA-GRM-CssCV on 12 and 7 test instances, respectively. In contrast, although both SaDE-SA-GRM-CVOB and SaDE-SA-GRM-CssCV identify feasible solutions in 20 independent runs on the CEC 2006 suite, they achieve significantly better

results than SaDE-SA-GRM on only three instances. These findings indicate that incorporating consensus across individual constraints together with objective improvement benefits performance in expensive constrained optimization. Specifically, enforcing consensus among all individual constraints is more effective than relying solely on overall constraint violation, which supports prioritizing the first for loop in the first step for candidate pre-selection. Figs. S.1 and S.2 (supplementary material) plot the convergence curves of objective function values and overall constraint violations against evolutionary generations for SaDE-SA-GRM on 12 representative test instances from CEC2006. Each subfigure is titled with the name of the corresponding test instance. As shown in Fig. S.1, SaDE-SA-GRM exhibits a very fast convergence speed in handling constraints during the early stages. The overall constraint violation can be reduced to near zero within 20 generations or less for most test instances. However, constraint satisfaction may experience stagnation in the middle stages, as observed in Figs. S.1(a), S.1(g), and S.1(k). In contrast, the convergence curves for objective function optimization exhibit significant fluctuations throughout the evolutionary process, as shown in Figs. S.2(a), S.2(g), and S.2(k). We attribute this behavior to the second step of the proposed infill sampling criterion, wherein the objective function value becomes decisive once consensus across all constraints has been established. When constraint violations of most individuals remain low, our developed infill sampling criterion tends to favor infeasible solutions that are predicted to yield smaller objective function values. Consequently, unlike the more stable curves observed for constraint satisfaction, the best objective function value may fluctuate significantly, jumping from a larger value to a smaller one across generations, resulting in a fluctuating convergence curve.

D. Influence of GRM

Gradient-based mutation has been widely shown to be effective for constrained optimization. However, few studies have leveraged its advantages for expensive constrained optimization. To assess the contribution of the proposed GRM, we remove it to obtain a variant, SaDE, and conduct a head-to-head comparison on the CEC2010 test set, which comprises more difficult and complex problems. Notably, in SaDE (which excludes GRM), the stagnation threshold that would otherwise invoke GRM is set to a very large value (1000), effectively disabling the mechanism and also serving as an abnormal-termination sentinel. Table S.VI (supplementary material) reports only the 17 test instances on which SaDE-SA-GRM and SaDE exhibit statistically significant differences in SR, AFV, or AOCV.

Among these 17 functions, 15 include at least one equality constraint. The ablation results indicate that SaDE-SA-GRM markedly improves the handling of equality constraints. First, SaDE-SA-GRM attains a 100% SR on 16 of the 17 instances and a 85% SR on C05₁₀, demonstrating consistent feasibility in 20 independent runs. In contrast, without GRM, SaDE fails to find feasible solutions on 9 instances; on the remaining 8 instances, its SR values are all below 40%.

Second, although SaDE's AOCV values are relatively small on many instances, transitioning from infeasible to feasible solutions remains difficult. For example, on C16₃₀ and C18₃₀ the overall constraint violations are on the order of 0.001—only one order of magnitude above the 0.0001 tolerance for equality constraints—yet SaDE achieves SRs of 20% and 0%, respectively. After introducing GRM, these small but consequential violations are reduced sufficiently for feasible solutions to be located consistently across all 20 runs. Third, as discussed in Section 1, equality constraints typically induce extremely narrow feasible regions. Of the 17 instances in Table S.VI, 15 contain equality constraints, on which SaDE-SA-GRM overwhelmingly outperforms SaDE.

Taken together, these findings demonstrate that GRM plays a significant role in enhancing constraint-handling capability for expensive equality-constrained optimization.

E. Further Discussion: Is GRM Cost-Effective under Stagnation Strategy?

GRM is proposed to address constraints that are difficult to optimize, such as equality constraints. In Subsection IV-D, GRM has been demonstrated to significantly enhance constraint-handling ability. However, it is well-known that gradient-based mutation employed in model-free EAs often requires a substantial number of FEs to numerically calculate the gradient, which makes gradient-based mutation impractical in expensive optimization scenarios. To analyze the cost-effectiveness of GRM in our proposed approach, we compute the average time (AT) of employing GRM and the SR value for each test instance. Table S.VII (supplementary material) presents the empirical results.

First, SaDE-SA-GRM achieves 100% SR values on most test instances from the CEC2006 and CEC2010 test sets. Meanwhile, the AT value remains very low for each test instance. For example, the maximum AT value is 3.10 for C16₃₀, indicating that, on average, only 93 expensive FEs are used by GRM to find a feasible solution. Considering the overall computational budget and constraint-handling ability, GRM consumes less than 10% of the expensive FEs while achieving a substantial improvement. Second, the AT values are zero for 29 out of 58 instances, indicating that GRM has not been activated during the evolutionary process. As we introduce stagnation in evolution, GRM is conditionally applied. Thus, if the global and local surrogate-driven evolution can handle the constraints effectively, the implementation of GRM is unnecessary.

Considering the investigation, it is evident that GRM improves constraint-handling ability, particularly for equality constraints. We believe that GRM is a cost-efficient method in our proposed approach for solving expensive constrained optimization problems.

F. Comparisons with Other SAEAs

The performance of SaDE-SA-GRM is compared with six state-of-the-art SAEAs: GLoSADE [20], DSI-C²oDE [10], MPMLS [34], SA-TSDE [22], SParEA [23], and eToSA-DE

[19]. Notably, GLoSADE, DSI-C²oDE, MPMLS, and SA-TSDE were originally developed for ECOPs with only inequality constraints, making them less suitable for the present evaluation involving equality constraints. Adapting algorithms designed exclusively for inequality-constrained ECOPs to handle equality constraints may yield biased comparisons. Moreover, few algorithms in the literature are explicitly designed to address equality constraints. Consequently, these four algorithms are primarily analyzed based on their performance on test instances containing only inequality constraints.

The parameter settings of the compared algorithms follow those recommended in their original publications, except that MaxFEs is uniformly set to 1000 for all algorithms on each test instance. The SR and AFV metrics achieved by the algorithms on the CEC2010 test suite are summarized in Table I. Additionally, the Wilcoxon rank-sum test at a significance level of 0.05 is conducted to statistically compare SaDE-SA-GRM with the other algorithms. Symbols “+”, “−”, and “≈” indicate that the compared algorithm performs significantly better, worse, or equivalently to SaDE-SA-GRM, respectively. For constrained optimization, feasible solutions are inherently preferred over infeasible ones, even when infeasible solutions yield lower objective function values. Thus, the SR metric is prioritized when comparing two algorithms, followed by the AFV metric.

As seen from Table I, SaDE-SA-GRM achieves a 100% SR on 29 test instances, comprising all 12 instances involving only inequality constraints, 9 out of 14 instances with only equality constraints, and 8 out of 10 instances containing both equality and inequality constraints. For comparison, SParEA and eToSA-DE achieve 100% SR on 6 and 25 test instances, respectively. Among algorithms originally designed exclusively for inequality-constrained ECOPs, GLoSADE, DSI-C²oDE, MPMLS, and SA-TSDE reach a 100% SR on 9, 10, 10, and 12 test instances, respectively. Excluding six extremely challenging test instances, where no algorithm successfully identifies feasible solutions, SaDE-SA-GRM fails on only one test instance across 20 independent runs, while eToSA-DE fails on 5 test instances. Furthermore, according to the Friedman test, SaDE-SA-GRM demonstrates the best overall performance by achieving the lowest rank score, followed by eToSA-DE. The statistical analysis, supported by the corresponding *p*-values, indicates that SaDE-SA-GRM significantly outperforms GLoSADE, DSI-C²oDE, MPMLS, SA-TSDE, and SParEA, and exhibits competitive performance compared to eToSA-DE.

As previously discussed, ECOPs can be categorized into three types based on their constraint structures: only inequality constraints, only equality constraints, and combined equality and inequality constraints. For test instances involving only inequality constraints, SaDE-SA-GRM significantly outperforms SParEA, GLoSADE, DSI-C²oDE, MPMLS, and SA-TSDE on 11, 10, 7, 7, and 7 test instances, respectively. Conversely, SaDE-SA-GRM performs significantly worse than these algorithms on 1, 2, 5, 5, and 5 test instances, respectively. When compared to eToSA-DE, which is specifically designed to handle all three constraint types, SaDE-SA-GRM achieves superior performance on 17 test instances, whereas eToSA-DE

TABLE I
EMPIRICAL RESULTS OF THE SIX COMPARED ALGORITHMS AND SADE-SA-GRM

Inst.	GLOSADE		DSI-C ² oDE		MPMLS		SA-TSDE		SParEA		eToSA-DE		SaDE-SA-GRM	
	SR	AFV	SR	AFV	SR	AFV	SR	AFV	SR	AFV	SR	AFV	SR	AFV
C01 ₁₀	1.00	-4.0761E-01 −	1.00	-4.3431E-01 −	1.00	-4.7216E-01 −	1.00	-5.4005E-01 +	1.00	-3.02E-01 −	1.00	-5.6863E-01 +	1.00	-5.0075E-01
C02 ₁₀	0.05	1.9103E+00 −	0.15	1.0167E+00 −	0.90	5.3761E-01 −	1.00	2.0256E+00 −	0.00	N.A. −	1.00	-2.3178E+00 +	1.00	-1.7193E+00
C03 ₁₀	0.00	N.A. −	0.05	2.1976E+12 −	0.05	8.0560E+12 −	0.00	N.A. −	0.00	N.A. −	1.00	8.62E+10 +	1.00	2.8734E+12
C04 ₁₀	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	1.00	-4.2524E+00 +	0.00	N.A.
C05 ₁₀	0.00	N.A. −	0.50	3.3242E+02 −	0.00	N.A. −	0.95	4.6189E+02 −	0.00	N.A. −	1.00	-2.5144E+02 +	1.00	-1.8217E+02
C06 ₁₀	0.00	N.A. −	0.60	3.3175E+02 −	0.00	N.A. −	0.35	5.1800E+02 −	0.00	N.A. −	1.00	-1.7074E+02 −	1.00	-4.4032E+02
C07 ₁₀	1.00	3.1996E+05 +	1.00	1.4983E+03 +	1.00	1.8820E+03 +	1.00	1.1952E+04 +	1.00	4.30E+04 +	1.00	6.38E+09 −	1.00	7.8413E+05
C08 ₁₀	0.00	N.A. −	1.00	7.4840E+04 +	1.00	1.1788E+04 +	1.00	8.4285E+05 −	1.00	1.70E+08 −	1.00	6.89E+09 −	1.00	5.7511E+05
C09 ₁₀	0.00	N.A. −	0.10	3.9104E+10 −	0.15	2.7920E+09 −	0.25	7.9641E+12 −	0.00	N.A. −	1.00	6.63E+11 +	1.00	1.9702E+12
C10 ₁₀	0.00	N.A. −	0.10	7.2371E+10 −	0.00	N.A. −	0.05	6.2655E+12 −	0.00	N.A. −	1.00	7.29E+11 +	1.00	1.4645E+12
C11 ₁₀	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A.
C12 ₁₀	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A.
C13 ₁₀	1.00	-5.4645E+01 +	1.00	-5.1442E-01 −	1.00	-5.7701E+01 +	1.00	-5.6047E+01 +	0.92	-2.29E+01 −	0.88	-2.2939E+01 −	1.00	-5.2728E+01
C14 ₁₀	1.00	2.8285E+10 −	1.00	6.8833E+13 −	1.00	6.5730E+09 −	1.00	6.6381E+13 −	0.84	1.1E+13 −	1.00	1.72E+14 −	1.00	3.3602E+08
C15 ₁₀	0.50	1.4737E+13 −	0.85	3.3103E+13 −	0.10	4.3403E+13 −	1.00	2.1803E+14 −	0.08	1.3E+12 −	0.08	1.77E+14 −	1.00	4.1176E+13
C16 ₁₀	0.00	N.A. −	1.00	1.0481E+00 −	0.65	9.7920E-01 −	1.00	1.0557E+00 −	0.00	N.A. −	1.00	1.6437E-01 +	1.00	5.8537E-01
C17 ₁₀	0.05	6.3841E-02 −	0.10	7.9555E+01 −	1.00	4.4600E+01 −	1.00	2.8494E+02 −	0.00	N.A. −	1.00	2.0511E-01 +	1.00	1.4457E+01
C18 ₁₀	1.00	3.5657E+03 −	1.00	4.9241E+03 −	1.00	2.1653E+03 −	1.00	9.0432E+03 −	0.00	N.A. −	1.00	1.0000E-06 +	1.00	7.8020E-01
C01 ₃₀	1.00	-2.2659E-01 −	1.00	-3.2181E-01 −	1.00	-3.2167E-01 −	1.00	-2.7346E-01 −	1.00	-2.12E-01 −	1.00	-2.0439E-01 −	1.00	-3.2521E-01
C02 ₃₀	0.10	2.2697E+00 −	0.00	N.A. −	0.90	4.0369E+00 −	0.90	4.1554E+00 −	0.00	N.A. −	1.00	1.9177E+00 −	1.00	1.0422E-01
C03 ₃₀	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A.
C04 ₃₀	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A.
C05 ₃₀	0.00	N.A. −	0.20	5.5638E+02 −	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	1.00	2.6014E+02 −	1.00	7.8396E+01
C06 ₃₀	0.00	N.A. −	0.20	3.6899E+02 −	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	1.00	2.8682E+02 −	1.00	-6.3557E+01
C07 ₃₀	1.00	5.3881E+07 −	1.00	4.1740E+05 +	1.00	2.0297E+06 +	1.00	2.1737E+06 +	1.00	2.80E+09 −	1.00	6.46E+10 −	1.00	1.6012E+07
C08 ₃₀	0.00	N.A. −	1.00	4.7846E+06 +	1.00	1.4224E+08 +	1.00	1.1560E+08 +	1.00	3.50E+09 −	1.00	8.54E+10 −	1.00	1.4505E+08
C09 ₃₀	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	1.00	1.07E+13 −	1.00	1.8516E+12
C10 ₃₀	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	1.00	1.16E+13 +	1.00	2.3801E+13
C11 ₃₀	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A.
C12 ₃₀	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A. ≈	0.00	N.A.
C13 ₃₀	1.00	-3.3626E+01 −	1.00	-5.2060E+01 +	1.00	-4.5892E+01 −	1.00	-5.0718E+01 −	0.84	-9.91E+00 −	0.96	-9.9175E+00 −	1.00	-5.1423E+01
C14 ₃₀	1.00	5.7306E+11 −	1.00	1.3251E+14 −	1.00	1.2689E+12 −	1.00	1.8844E+14 −	0.92	6.7E+13 −	1.00	4.15E+14 −	1.00	4.9078E+10
C15 ₃₀	0.75	2.2710E+12 −	0.90	3.0351E+14 −	0.45	1.2271E+14 −	1.00	1.1035E+15 −	0.20	1.4E+14 −	0.24	6.30E+14 −	1.00	3.7898E+14
C16 ₃₀	0.00	N.A. −	1.00	1.1214E+00 −	0.40	1.0447E+00 −	0.05	1.1330E+00 −	0.00	N.A. −	0.76	1.0217E+00 −	1.00	9.3326E-01
C17 ₃₀	0.00	N.A. −	0.25	5.8237E+02 −	0.80	6.3969E+02 −	0.00	N.A. −	0.00	N.A. −	1.00	4.4268E+01 +	1.00	9.2735E+02
C18 ₃₀	0.40	1.6444E-04 −	0.35	2.3748E+04 −	1.00	1.5860E+04 −	0.95	4.0749E+04 −	0.00	N.A. −	1.00	4.5972E+01 +	1.00	2.1776E+02
Cmp.	Wins: 2; Lose: 27		Wins: 5; Lose: 24		Wins: 5; Lose: 24		Wins: 6; Lose: 23		Wins: 1; Lose: 28		Wins: 13; Lose: 17		NaN	
Score	104.23		89.88		93.75		86.36		116.75		63.70		41.80	
p-value	5.93E-07		6.32E-05		8.22E-05		1.40E-05		2.60E-08		8.18E-02		NaN	

obtains better results on 14 test instances. Overall, SaDE-SA-GRM demonstrates a clear advantage in effectively handling all three types of instances from CEC2010 test suite, primarily due to the following reasons:

- 1) SaDE-SA-GRM integrates gradient information into its local search procedure, enhancing its ability to locate feasible solutions. Moreover, the global and local surrogate-driven evolution processes provide sufficient generations to accurately optimize the surrogate models of constraints. With well-constructed surrogate models, equality constraints can be addressed more effectively, similar to conventional model-free algorithms, thereby enhancing SaDE-SA-GRM's performance in handling equality constraints.
- 2) SaDE-SA-GRM exhibits stable and robust performance across all three types of ECOPs. This stability can be attributed to the proposed infill sampling criterion and the developed stagnation mechanism within the evolutionary

strategy. The stagnation mechanism provides SaDE-SA-GRM with fault-tolerant generations, facilitating exploration of diverse search regions and reducing premature convergence that typically arises from frequent local searches. Furthermore, the consensus between objective function optimization and constraint satisfaction effectively mitigates search biases caused by specific infeasible solutions with extremely small constraint violations.

Additionally, a run-time performance comparison is provided in Table S.VIII (supplementary material), highlighting the computational complexity of SaDE-SA-GRM. Theoretically, the time complexity of the RBFN is $O(I \cdot S \ln S \cdot n)$, where I is the number of training iterations, S denotes the size of the training data, and n represents the dimensionality of the problem. However, actual computational performance greatly depends on the evolutionary strategy employed, potentially extending the training process. As indicated in Table S.VIII, the run-time of SaDE-SA-GRM remains moderate for solving the

CEC2010 test instances. It is faster than SParEA and eToSA-DE on most test instances but exhibits slower performance compared to GLoSADE, DSI-C²oDE, MPMLS, and SA-TSDE. Two primary factors contribute to SaDE-SA-GRM's increased computational demands: 1) the introduced stagnation mechanism occasionally discards newly generated offspring without conducting expensive evaluations, leading SaDE-SA-GRM to repeatedly retrain the RBFN surrogate models; 2) SaDE-SA-GRM separately constructs surrogate models for the objective function and each individual constraint. For instance, when addressing the test instances C07₁₀ and C08₁₀, Table S.VI (supplementary material) indicates that GRM is not activated for constraint handling, causing exploration and exploitation of feasible solutions to rely exclusively on surrogate-driven evolution. Consequently, SaDE-SA-GRM incurs notably longer computational times compared to GLoSADE, DSI-C²oDE, MPMLS, SA-TSDE, and eToSA-DE. However, this strategy provides SaDE-SA-GRM more opportunities to identify high-quality candidate solutions, ultimately resulting in significantly improved feasible solutions for the C07₁₀ and C08₁₀ test instances.

Based on these comparative analyses, it can be concluded that SaDE-SA-GRM is highly competitive with state-of-the-art SAEAs specifically designed for inequality-constrained ECOPs, and notably more applicable for effectively solving ECOPs encompassing all three types of constraints.

V. APPLICATION IN REAL-WORLD PROBLEMS

In this section, SaDE-SA-GRM is further applied to practical ECOPs, specifically pulse-width modulation (PWM) optimization [44], [45]. PWM optimization constitutes a complex and computationally expensive task, primarily due to the requirement of time-intensive PWM signal simulations performed in MATLAB to assess solution quality [46]. PWM effectively reduces switching frequency without increasing harmonic distortion, thereby minimizing switching losses and enhancing inverter performance. Over a fundamental period, switching angles must be calculated to reduce current distortion effectively. For multilevel inverters, the PWM optimization can be formulated as a scalable COP as follows:

$$\begin{aligned}
 &\text{minimize} && f_d(\mathbf{x}), && \mathbf{x} = (x_1, \dots, x_n), \\
 &\text{subject to} && f_s^i(\mathbf{x}) > 0, && i = 1, \dots, n-1, \\
 &&& f_c(\mathbf{x}) = 0, \\
 &\text{with bounds:} && 0 < x_i < \frac{\pi}{2}, && i = 1, \dots, n,
 \end{aligned} \tag{17}$$

where n denotes the pulse number, $f_d(\cdot)$ represents the total harmonic distortion, $f_s^i(\cdot)$ denotes constraints related to amplitude changes between harmonics, and $f_c(\cdot)$ indicates the constraint enforcing the elimination of a specified fundamental harmonic component. Signal simulations conducted in MATLAB for six-level inverters require approximately 15 s to 30 s per simulation on our workstation.

The proposed SaDE-SA-GRM approach is compared against three state-of-the-art SAEAs: DSI-C²oDE, MPMLS, and SA-TSDE, in this real-world application. Considering the computational complexity of simulations, the *MaxFEs* is limited to 1000, and each algorithm is independently executed

TABLE II
STATISTICAL RESULTS ACHIEVED BY ALGORITHMS ON PWM OPTIMIZATION PROBLEMS

Level & Pulse	DSI-C ² oDE		MPMLS		SA-TSDE		SaDE-SA-GRM	
	SR	AFV	SR	AFV	SR	AFV	SR	AFV
3 & 25	1.0	0.0572	1.0	0.2036	1.0	0.1088	1.0	0.1624
5 & 25	1.0	0.2251	0.6	0.2479	1.0	0.4209	1.0	0.2116
7 & 25	1.0	0.1469	0.6	0.3180	1.0	0.4686	1.0	0.1418
9 & 30	1.0	0.3353	0.5	0.4749	1.0	0.7156	1.0	0.1748
11 & 30	0.9	0.3155	0.3	0.4012	1.0	0.2787	1.0	0.0832
13 & 30	0.5	0.2693	0.2	0.3276	1.0	0.3415	1.0	0.0706

ten times. Parameter settings for DSI-C²oDE, MPMLS, and SA-TSDE follow those recommended in their original publications, while the configuration of SaDE-SA-GRM aligns with the specifications provided in Section IV-A. Statistical results of these comparative experiments are presented in Table II.

From Table II, it is evident that with increasing inverter levels, DSI-C²oDE and MPMLS fail to consistently handle constraints across ten independent runs. Conversely, SA-TSDE and SaDE-SA-GRM achieve a 100% SR for all tested inverter levels. Furthermore, SaDE-SA-GRM demonstrates superior overall performance in terms of the AFV compared to SA-TSDE. Hence, SaDE-SA-GRM exhibits competitive performance relative to DSI-C²oDE, MPMLS, and SA-TSDE in addressing complex real-world PWM optimization problems.

VI. CONCLUSION

In this work, we propose a SAEA, referred to as SaDE-SA-GRM, to solve ECOPs, targeting all three types of constraints within a strict budget of 1000 expensive FEs. The proposed SaDE-SA-GRM incorporates three key evolutionary strategies: a stagnation strategy for managing surrogate-driven evolution and model-free optimization, a consensus-aware infill sampling criterion, and a cost-effective GRM. During the evolutionary process, the stagnation strategy is designed to balance the surrogate-driven evolution for effective exploration and the model-free GRM for economical exploitation of feasible regions. The proposed infill sampling criterion, developed based on the concept of consensus, considers both objective function optimization and constraint satisfaction. Finally, GRM is developed gradient-based mutation from our proposed reused strategy to address complex and challenging constraints in a economical manner.

To evaluate the effectiveness of SaDE-SA-GRM, we investigate parameter sensitivity, the influence of each evolutionary strategy, and the cost-efficiency of GRM under the stagnation strategy. The overall performance is validated using 58 expensive test instances from the CEC2006 and CEC2010 test sets, encompassing three types of ECOPs: 1) those with only inequality constraints, 2) those with only equality constraints, and 3) those with both equality and inequality constraints. Empirical results demonstrate that SaDE-SA-GRM achieves superior or highly competitive performance compared to six state-of-the-art SAEAs.

For future work, we aim to eliminate explicit gradient computation and instead explore alternative methods for extracting gradient information for mutation. Additionally, extend-

ing SaDE-SA-GRM to address expensive constrained multi-objective optimization problems presents a promising research direction.

VII. ACKNOWLEDGEMENT

The authors would like to thank Dr. G. Li for providing the source code of MPMLS and Dr. Y. Liu for providing the source code of SA-TSDE. They would also like to thank Prof. Y. Wang for making the source code of GLoSADE publicly available and Prof. H. Wang for providing public access to the source code of DSI-C²oDE.

REFERENCES

- [1] S. Venkatraman and G. Yen, "A generic framework for constrained optimization using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 4, pp. 424–435, 2005.
- [2] Y. Liu, J. Liu, J. Ding, S. Yang, and Y. Jin, "A surrogate-assisted differential evolution with knowledge transfer for expensive incremental optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 4, pp. 1039–1053, 2024.
- [3] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2010.
- [4] Z. Song, H. Wang, B. Xue, M. Zhang, and Y. Jin, "Balancing objective optimization and constraint satisfaction in expensive constrained evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 5, pp. 1286–1300, 2024.
- [5] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
- [6] R. Mallipeddi and P. N. Suganthan, "Ensemble of constraint handling techniques," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 561–579, 2010.
- [7] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 658–675, 2006.
- [8] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2014.
- [9] F. Li, X. Cai, L. Gao, and W. Shen, "A surrogate-assisted multiswarm optimization algorithm for high-dimensional computationally expensive problems," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1390–1402, 2021.
- [10] Z. Song, H. Wang, and Y. Jin, "A surrogate-assisted evolutionary framework with regions of interests-based data selection for expensive constrained optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 10, pp. 6268–6280, 2023.
- [11] S. D. Handoko, C. K. Kwoh, and Y.-S. Ong, "Feasibility structure modeling: An effective chapterone for constrained memetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 740–758, 2010.
- [12] F.-F. Wei, W.-N. Chen, Q. Yang, J. Deng, X.-N. Luo, H. Jin, and J. Zhang, "A classifier-assisted level-based learning swarm optimizer for expensive optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 2, pp. 219–233, 2021.
- [13] G. Li, Z. Wang, and M. Gong, "Expensive optimization via surrogate-assisted and model-free evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 5, pp. 2758–2769, 2023.
- [14] F. Li, L. Gao, and W. Shen, "Surrogate-assisted multi-objective evolutionary optimization with Pareto front model-based local search method," *IEEE Transactions on Cybernetics*, vol. 54, no. 1, pp. 173–186, 2024.
- [15] A. Zhou, J. Sun, and Q. Zhang, "An estimation of distribution algorithm with cheap and expensive local search methods," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 807–822, 2015.
- [16] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.
- [17] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2, pp. 311–338, 2000.
- [18] C. Hu, S. Zeng, and C. Li, "An uncertainty measure for prediction of non-Gaussian process surrogates," *Evolutionary Computation*, vol. 31, no. 1, pp. 53–71, 03 2023.
- [19] F.-F. Wei, W.-N. Chen, W. Mao, X.-M. Hu, and J. Zhang, "An efficient two-stage surrogate-assisted differential evolution for expensive inequality constrained optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 12, pp. 7769–7782, 2023.
- [20] 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 Transactions on Cybernetics*, vol. 49, no. 5, pp. 1642–1656, 2019.
- [21] J.-Y. Li, Z.-H. Zhan, and J. Zhang, "Evolutionary computation for expensive optimization: A survey," *Machine Intelligence Research*, vol. 19, no. 1, pp. 3–23, 2022.
- [22] Y. Liu, J. Liu, Y. Jin, F. Li, and T. Zheng, "A surrogate-assisted two-stage differential evolution for expensive constrained optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 715–730, 2023.
- [23] K. H. Rahi, H. K. Singh, and T. Ray, "Partial evaluation strategies for expensive evolutionary constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 6, pp. 1103–1117, 2021.
- [24] C. C. Tutum, S. Sayed, and R. Mäkeläinen, "Surrogate-based evolutionary optimization for friction stir welding," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 387–394.
- [25] Z. Yang, H. Qiu, L. Gao, X. Cai, C. Jiang, and L. Chen, "Surrogate-assisted classification-collaboration differential evolution for expensive constrained optimization problems," *Information Sciences*, vol. 508, pp. 50–63, 2020.
- [26] R. Jiao, B. Xue, and M. Zhang, "Investigating the correlation amongst the objective and constraints in Gaussian process-assisted highly constrained expensive optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 872–885, 2022.
- [27] C. Hu, S. Zeng, C. Li, and F. Zhao, "On nonstationary Gaussian process model for solving data-driven optimization problems," *IEEE Transactions on Cybernetics*, vol. 53, no. 4, pp. 2440–2453, 2023.
- [28] J. Liu, Y. Wang, G. Sun, and T. Pang, "Constrained evolutionary bayesian optimization for expensive constrained optimization problems with inequality constraints," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 3, pp. 2009–2021, 2025.
- [29] Y. Wang, J.-P. Li, X. Xue, and B.-C. Wang, "Utilizing the correlation between constraints and objective function for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 29–43, 2020.
- [30] P. Chootinan and A. Chen, "Constraint handling in genetic algorithms using a gradient-based repair method," *Computers & Operations Research*, vol. 33, no. 8, pp. 2263–2281, 2006.
- [31] Y. Wang, B.-C. Wang, H.-X. Li, and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2938–2952, 2016.
- [32] D. Lowe and D. Broomhead, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, no. 3, pp. 321–355, 1988.
- [33] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2013.
- [34] G. Li and Q. Zhang, "Multiple penalties and multiple local surrogates for expensive constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 769–778, 2021.
- [35] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [36] B.-C. Wang, H.-X. Li, J.-P. Li, and Y. Wang, "Composite differential evolution for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1482–1495, 2019.
- [37] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. C. Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," *Journal of Applied Mechanics*, vol. 41, no. 8, pp. 8–31, 2006.
- [38] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization,"

- IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [39] N. M. Hamza, D. L. Essam, and R. A. Sarker, “Constraint consensus mutation-based differential evolution for constrained optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 447–459, 2016.
 - [40] J.-Y. Ji, Z. Tan, S. Zeng, E. W. K. See-To, and M.-L. Wong, “A surrogate-assisted evolutionary algorithm for seeking multiple solutions of expensive multimodal optimization problems,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 1, pp. 377–388, 2024.
 - [41] R. L. Iman, “Latin hypercube sampling,” in *John Wiley & Sons, Ltd*, 2008.
 - [42] R. Mallipeddi and P. N. Suganthan, “Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization,” *Nanyang Technological University, Singapore*, vol. 24, p. 910, 2010.
 - [43] 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 and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
 - [44] A. Edpuganti and A. K. Rathore, “Fundamental switching frequency optimal pulsewidth modulation of medium-voltage cascaded seven-level inverter,” *IEEE Transactions on Industry Applications*, vol. 51, no. 4, pp. 3485–3492, 2015.
 - [45] —, “Optimal pulsewidth modulation for common-mode voltage elimination scheme of medium-voltage modular multilevel converter-fed open-end stator winding induction motor drives,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 848–856, 2017.
 - [46] M. B. Martin, “Pulse-width modulation (PWM) signal simulation with user-defined sampling frequency and modulating frequency using MATLAB,” in *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS)*, 2021, pp. 32–35.

Supplementary Material for “Surrogate-Assisted Evolutionary Algorithm for Expensive Optimization with Equality and Inequality Constraints”

LIST OF TABLES

S.I	PROPERTIES OF CEC2006 TEST PROBLEMS	3
S.II	PROPERTIES OF CEC2010 TEST PROBLEMS	3
S.III	Statistical Results for the Sensitivity Analysis of the Parameter Tm	4
S.IV	Statistical Results for the Sensitivity Analysis of the Parameter s_{num}	6
S.V	Statistical Results for the Influence Analysis of the Proposed Infill Sampling Criterion	7
S.VI	Statistical Results for the Influence Analysis of GRM	8
S.VII	Statistical Results for the Cost-Effectiveness Analysis of GRM	9
S.VIII	Run-Time Results of the Six Compared Algorithms and SaDE-SA-GRM	10

LIST OF FIGURES

S.1	Evolution of the overall constraint violations of SaDE-SA-GRM on 12 test instances from CEC2006.	5
S.2	Evolution of the objective function values of SaDE-SA-GRM on 12 test instances from CEC2006.	5
S.3	A schematic illustration of the GRM process applied to the given example. The black point $\mathbf{x}$ represents the initial infeasible solution. The gray-shaded region comprises solutions satisfying constraints g_1 and g_2 but violating the equality constraint h_1 , whereas the blue line denotes solutions satisfying h_1 but violating g_1 and g_2 . The red line indicates the gradient direction of the constraint h_1 at the solution $\mathbf{x}$	13

Algorithm S.1: SaDE-SA-GRM-CVOB

Input:

- the original population: $P = \{\mathbf{p}_1, \dots, \mathbf{p}_{NP}\}$;
- the newly evolved population: $Q = \{\mathbf{q}_1, \dots, \mathbf{q}_{NP}\}$;
- the associated parameters: c_g and c_h ;

Set $S = \{\}$;**for** $i = 1$ **to** NP **do** **if** $CV(\mathbf{q}_i) \leq CV(\mathbf{p}_i)$ **then** $S = S \cup \{\mathbf{q}_i\}$;**if** S is not empty **then** Perform an expensive evaluation of the individual with the minimum objective value in S (denoted $\mathbf{q}$); Identify the individual in P with the maximum overall constraint violation (denoted $\mathbf{p}$); **if** $\mathbf{q}$ is preferred to $\mathbf{p}$ according to the feasibility rule **then** Replace $\mathbf{p}$ with $\mathbf{q}$ and output P ; **else** Output stagnation;**else** Output stagnation;

Algorithm S.2: SaDE-SA-GRM-CssCV

Input:

- the original population: $P = \{\mathbf{p}_1, \dots, \mathbf{p}_{NP}\}$;
- the newly evolved population: $Q = \{\mathbf{q}_1, \dots, \mathbf{q}_{NP}\}$;
- the associated parameters: c_g and c_h ;

Set $S = \{\}$;**for** $i = 1$ **to** NP **do** **if** $g_j(\mathbf{q}_i) \leq g_j(\mathbf{p}_i), j = 1, \dots, c_g$ and $h_t(\mathbf{q}_i) \leq h_t(\mathbf{p}_i), t = 1, \dots, c_h$ **then** $S = S \cup \{\mathbf{q}_i\}$;**if** S is empty **then** **for** $i = 1$ **to** NP **do** **if** $CV(\mathbf{q}_i) \leq CV(\mathbf{p}_i)$ **then** $S = S \cup \{\mathbf{q}_i\}$;**if** S is not empty **then** Identify the best individual in S with the minimum overall constraint violation (denoted $\mathbf{q}$); Perform an expensive evaluation of the individual $\mathbf{q}$; Identify the individual in P with the maximum overall constraint violation (denoted $\mathbf{p}$); **if** $\mathbf{q}$ is preferred to $\mathbf{p}$ according to the feasibility rule **then** Replace $\mathbf{p}$ with $\mathbf{q}$ and output P ; **else** Output stagnation;**else** Output stagnation;

TABLE S.I
PROPERTIES OF CEC2006 TEST PROBLEMS

Inst.	n	Objective Type	Number of Constraints		Feasibility Region $\rho(\%)$
			c_h	c_g	
G01	13	Quadratic	0	9	0.0111
G02	20	Nonlinear	0	2	99.8474
G03	10	Polynomial	1	0	0.0000
G04	5	Quadratic	0	6	52.1230
G05	4	Cubic	3	2	0.0000
G06	2	Cubic	0	2	0.0066
G07	10	Quadratic	0	8	0.0003
G08	2	Nonlinear	0	2	0.8560
G09	7	Polynomial	0	4	0.5121
G10	8	Linear	0	6	0.0010
G11	2	Quadratic	1	0	0.0000
G12	3	Quadratic	0	1	4.7713
G13	5	Nonlinear	3	0	0.0000
G14	10	Nonlinear	3	0	0.0000
G15	3	Quadratic	2	0	0.0000
G16	5	Nonlinear	0	38	0.0204
G17	6	Nonlinear	4	0	0.0000
G18	9	Quadratic	0	13	0.0000
G19	15	Nonlinear	0	5	33.4761
G21	7	Linear	5	1	0.0000
G23	9	Linear	4	2	0.0000
G24	2	Linear	0	2	79.6556

TABLE S.II
PROPERTIES OF CEC2010 TEST PROBLEMS

Inst.	Objective Type	Number of Constraints / Type of constraint		Feasibility Region $\rho(\%)$	
		c_h	c_g	$n = 10$	$n = 30$
C01	Non Separable	0	2/Non Separable	99.7689	100.0000
C02	Separable	1/Separable	2/Separable	0.0000	0.0000
C03	Non Separable	1/Non Separable	0	0.0000	0.0000
C04	Separable	2/Non Separable, 2/Separable	0	0.0000	0.0000
C05	Separable	2/Separable	0	0.0000	0.0000
C06	Separable	2/Rotated	0	0.0000	0.0000
C07	Non Separable	0	1/Separable	50.5123	50.3725
C08	Non Separable	0	1/Rotated	37.9512	37.5278
C09	Non Separable	1/Separable	0	0.0000	0.0000
C10	Non Separable	1/Rotated	0	0.0000	0.0000
C11	Rotated	1/Non Separable	0	0.0000	0.0000
C12	Separable	1/Non Separable	1/Separable	0.0000	0.0000
C13	Separable	0	2/Separable, 1/Non Separable	0.0000	0.0000
C14	Non Separable	0	3/Separable	0.3112	0.6123
C15	Non Separable	0	3/Rotated	0.3210	0.6023
C16	Non Separable	2/Separable	2/Separable, 1/Non Separable	0.0000	0.0000
C17	Non Separable	1/Separable	2/Non Separable	0.0000	0.0000
C18	Non Separable	1/Separable	1/Separable	0.0010	0.0000

TABLE S.III
STATISTICAL RESULTS FOR THE SENSITIVITY ANALYSIS OF THE PARAMETER Tm

Inst.	$Tm = 1$		$Tm = 20$		$Tm = 50$		$Tm = 100$		$Tm = 200$		$Tm = 300$	
	SR	AFV	SR	AFV	SR	AFV	SR	AFV	SR	AFV	SR	AFV
G01	1.00	-14.8856	1.00	-15.0000	1.00	-15.0000	1.00	-15.0000	1.00	-15.0000	1.00	-15.0000
G02	1.00	-0.4878	1.00	-0.4730	1.00	-0.4979	1.00	-0.4082	1.00	-0.4359	1.00	-0.3870
G03	1.00	-0.4377	1.00	-0.7999	1.00	-0.8012	1.00	-0.6855	1.00	-0.6002	1.00	-0.6635
G04	1.00	-30665.5385	1.00	-30665.5387	1.00	-30665.5387	1.00	-30665.5387	1.00	-30665.5387	1.00	-30665.5387
G05	0.00	5031.5664	1.00	5126.4969	1.00	5126.4967	1.00	5126.4967	1.00	5126.4967	1.00	5126.4967
G06	0.45	-5387.8185	1.00	-6961.8139	1.00	-6961.8138	1.00	-6961.8138	1.00	-6961.8139	1.00	-6961.8139
G07	1.00	24.3600	1.00	24.3063	1.00	24.3064	1.00	24.3066	1.00	24.3068	1.00	24.3063
G08	0.80	-0.0584	1.00	-0.0686	1.00	-0.0614	1.00	-0.0414	1.00	-0.0616	1.00	-0.0546
G09	1.00	808.5231	1.00	892.8880	1.00	921.2805	1.00	912.6362	1.00	825.3350	1.00	817.6810
G10	0.90	7001.3782	1.00	7190.0313	1.00	7049.5028	1.00	7049.3564	1.00	7049.7524	1.00	7049.5106
G11	0.10	0.8162	1.00	0.7499	1.00	0.7499	1.00	0.7499	1.00	0.7499	1.00	0.7499
G12	1.00	-0.9972	1.00	-0.9856	1.00	-0.9619	1.00	-0.9689	1.00	-0.9597	1.00	-0.9714
G13	0.75	0.9176	1.00	1.2530	1.00	1.1920	1.00	1.4615	1.00	1.4820	1.00	0.8213
G14	0.00	-106.8524	1.00	-47.7636	1.00	-47.7649	1.00	-47.7649	1.00	-47.7649	1.00	-47.7648
G15	0.00	965.7521	1.00	961.7150	1.00	961.7150	1.00	961.7150	1.00	961.7150	1.00	961.7150
G16	0.90	-1.7420	1.00	-1.8908	1.00	-1.9050	1.00	-1.9052	1.00	-1.9052	1.00	-1.9052
G17	0.00	8753.4847	1.00	8856.9857	1.00	8859.1159	1.00	8867.4774	1.00	8878.6355	1.00	8882.7026
G18	1.00	-0.7737	1.00	-0.8660	1.00	-0.8404	1.00	-0.8447	1.00	-0.8278	1.00	-0.7672
G19	1.00	42.1523	1.00	44.3972	1.00	48.3852	1.00	43.5966	1.00	46.6711	1.00	41.0384
G21	0.00	37.4318	0.00	39.7798	0.00	144.8118	0.70	236.0085	1.00	262.6105	1.00	249.2311
G23	0.00	-1235.0056	0.60	-399.6546	1.00	-400.0551	1.00	-400.0551	1.00	-400.0551	1.00	-400.0551
G24	1.00	-5.5080	1.00	-5.5080	1.00	-5.5080	1.00	-5.5080	1.00	-5.5080	1.00	-5.5080
Average	0.6318	-670.6238	0.9364	-680.0800	0.9545	-680.1451	0.9864	-676.2150	1.00	-678.3053	1.00	-679.3717

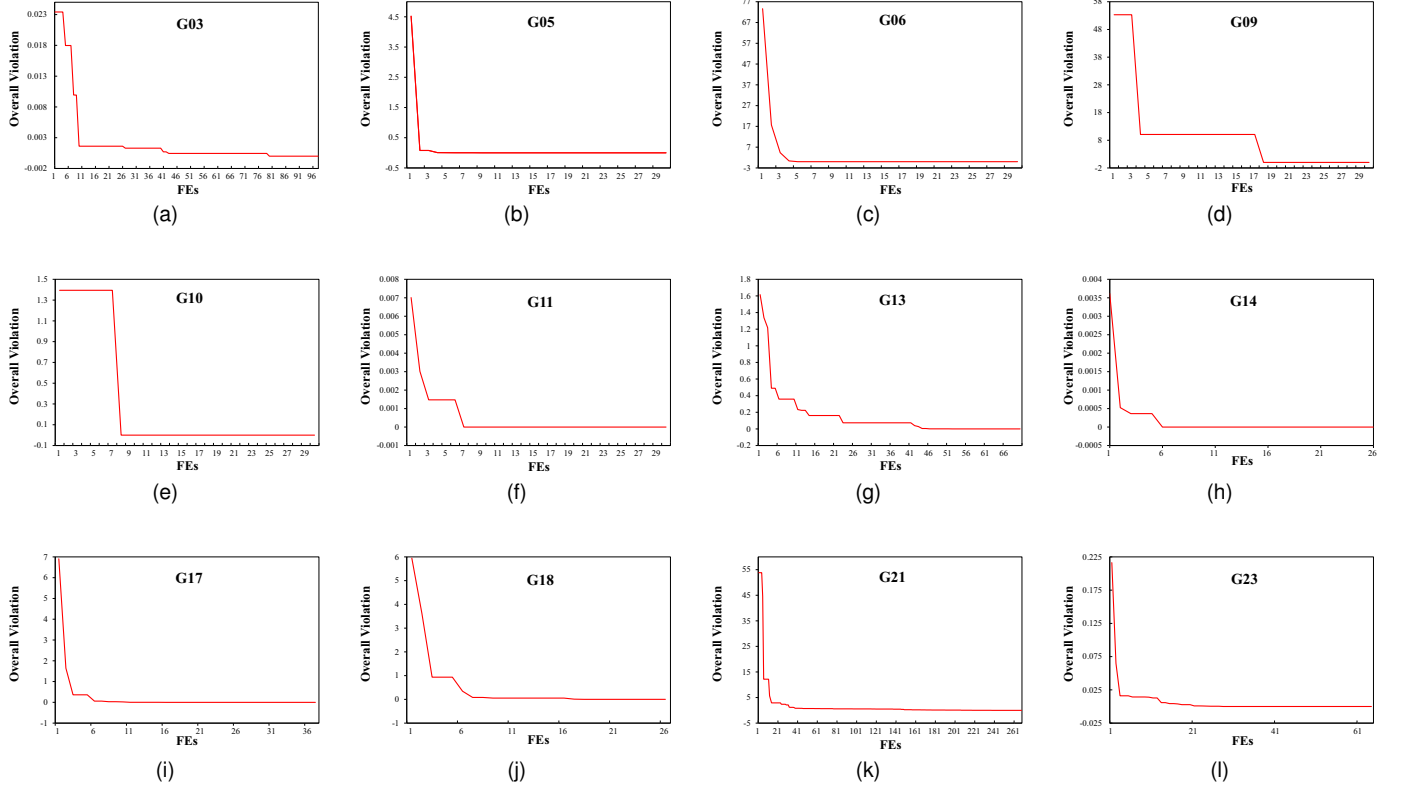


Fig. S.1. Evolution of the overall constraint violations of SaDE-SA-GRM on 12 test instances from CEC2006.

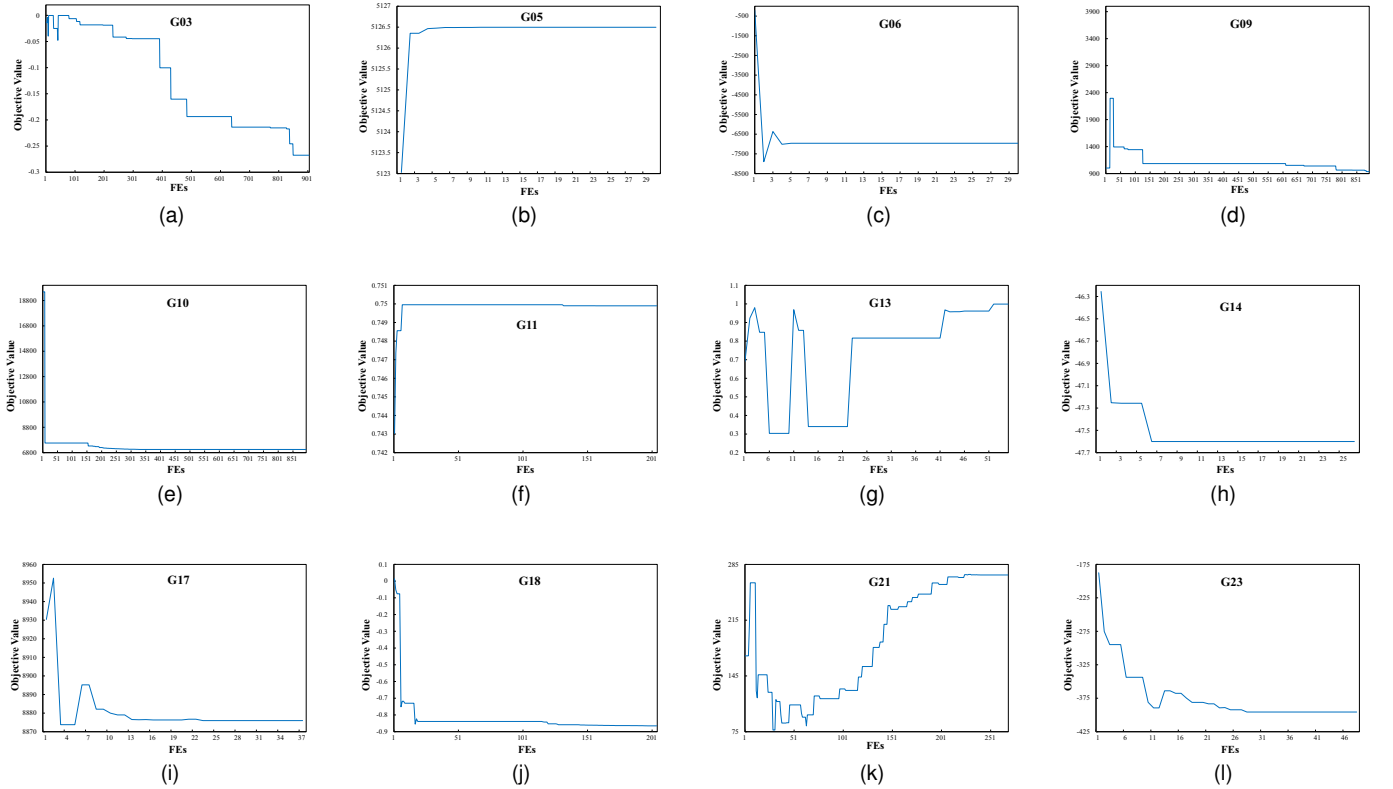


Fig. S.2. Evolution of the objective function values of SaDE-SA-GRM on 12 test instances from CEC2006.

TABLE S.IV
STATISTICAL RESULTS FOR THE SENSITIVITY ANALYSIS OF THE PARAMETER s_{num}

Inst.	$s_{num} = 1$		$s_{num} = 5$		$s_{num} = 10$	
	SR	AFV	SR	AFV	SR	AFV
G01	1.00	-15.0000	1.00	-15.0000	1.00	-15.0000
G02	1.00	-0.4266	1.00	-0.4359	1.00	-0.4312
G03	1.00	-0.8290	1.00	-0.6002	1.00	-0.7980
G04	1.00	-30665.5387	1.00	-30665.5387	1.00	-30665.5387
G05	1.00	5126.4967	1.00	5126.4967	1.00	5126.4967
G06	1.00	-6961.8139	1.00	-6961.8139	1.00	-6961.8138
G07	1.00	24.3071	1.00	24.3068	1.00	24.3075
G08	1.00	-0.0452	1.00	-0.0616	1.00	-0.0756
G09	1.00	841.8100	1.00	825.3350	1.00	837.8380
G10	1.00	7049.6409	1.00	7049.7524	1.00	7050.2570
G11	1.00	0.7499	1.00	0.7499	1.00	0.7499
G12	1.00	-0.9540	1.00	-0.9597	1.00	-0.9615
G13	1.00	1.3927	1.00	1.4820	1.00	0.9621
G14	1.00	-47.7648	1.00	-47.7649	1.00	-47.7648
G15	1.00	961.7150	1.00	961.7150	1.00	961.7150
G16	1.00	-1.9050	1.00	-1.9052	1.00	-1.9051
G17	1.00	8887.7479	1.00	8878.6355	1.00	8865.7985
G18	1.00	-0.7339	1.00	-0.8278	1.00	-0.8087
G19	1.00	44.6715	1.00	46.6711	1.00	50.6656
G21	1.00	266.5107	1.00	262.6105	1.00	263.1270
G23	1.00	-400.0545	1.00	-400.0551	1.00	-400.0518
G24	1.00	-5.5080	1.00	-5.5080	1.00	-5.5080
Average	1.00	-677.0696	1.00	-678.3053	1.00	-678.1245

TABLE S.V
STATISTICAL RESULTS FOR THE INFLUENCE ANALYSIS OF THE PROPOSED INFILL SAMPLING CRITERION

Inst.	SaDE-SA-GRM-CVOB		SaDE-SA-GRM-CssCV		SaDE-SA-GRM	
	SR	AFV	SR	AFV	SR	AFV
G01	1.00	-14.7656 —	1.00	-15.0000 $\approx$	1.00	-15.0000
G02	1.00	-0.3791 —	1.00	-0.4316 —	1.00	-0.4359
G03	1.00	-0.9042 +	1.00	-0.7920 +	1.00	-0.6002
G04	1.00	-30665.5387 $\approx$	1.00	-30665.5387 $\approx$	1.00	-30665.5387
G05	1.00	5126.4981 —	1.00	5126.4967 $\approx$	1.00	5126.4967
G06	1.00	-6961.8137 $\approx$	1.00	-6961.8139 $\approx$	1.00	-6961.8139
G07	1.00	24.3615 —	1.00	24.3064 $\approx$	1.00	24.3068
G08	1.00	-0.0622 $\approx$	1.00	-0.0616 $\approx$	1.00	-0.0616
G09	1.00	819.3198 +	1.00	1000.0135 —	1.00	825.3350
G10	1.00	7105.3928 —	1.00	7050.3495 —	1.00	7049.7524
G11	1.00	0.7499 $\approx$	1.00	0.7499 $\approx$	1.00	0.7499
G12	1.00	-0.9582 —	1.00	-0.9651 +	1.00	-0.9597
G13	1.00	0.7372 +	1.00	1.6301 —	1.00	1.4820
G14	1.00	-47.0485 —	1.00	-47.7648 $\approx$	1.00	-47.7649
G15	1.00	961.7150 $\approx$	1.00	961.7150 $\approx$	1.00	961.7150
G16	1.00	-1.9046 $\approx$	1.00	-1.9052 $\approx$	1.00	-1.9052
G17	1.00	8881.7646 —	1.00	8879.3787 —	1.00	8878.6355
G18	1.00	-0.7352 —	1.00	-0.8469 +	1.00	-0.8278
G19	1.00	53.3500 —	1.00	46.6744 —	1.00	46.6711
G21	1.00	289.2162 —	1.00	277.7633 —	1.00	262.6105
G23	1.00	-371.6532 —	1.00	-400.0550 $\approx$	1.00	-400.0551
G24	1.00	-5.5080 $\approx$	1.00	-5.5080 $\approx$	1.00	-5.5080
Cmp.	Wins: 3; Lose: 12; Ties: 5		Wins: 3; Lose: 7; Ties 12		NaN	
Score	45.04		29.15		26.29	
<i>p</i> -value	2.96E-02		1.54E-01		NaN	

TABLE S.VI
STATISTICAL RESULTS FOR THE INFLUENCE ANALYSIS OF GRM

Inst.	SaDE-SA-GRM			SaDE		
	SR	AFV	AOCV	SR	AFV	AOCV
C02 ₁₀	1.00	-1.7193E+00	0.0	0.30	-1.5919	4.5914E-04
C02 ₃₀	1.00	1.0422E-01	0.0	0.15	1.7641	2.7089E-04
C03 ₁₀	1.00	2.8734E+12	0.0	0.25	6.6906E+12	2.5597E-03
C05 ₁₀	0.85	-1.7967E+02	9.96E-04	0.30	-4.0348E+02	5.9454E-01
C05 ₃₀	1.00	7.8396E+01	0.0	0.00	N/A	1.0854E+00
C06 ₃₀	1.00	-6.3557E+01	0.0	0.00	N/A	4.2152E-01
C09 ₁₀	1.00	1.9702E+12	0.0	0.00	N/A	2.5546E-01
C09 ₃₀	1.00	1.8516E+14	0.0	0.00	N/A	1.0793E+00
C10 ₁₀	1.00	1.4645E+12	0.0	0.00	N/A	3.2514E-01
C10 ₃₀	1.00	2.3801E+13	0.0	0.00	N/A	8.2775E-01
C15 ₁₀	1.00	4.1176E+13	0.0	0.35	3.2944E+13	4.7604E+01
C15 ₃₀	1.00	3.7898E+14	0.0	0.10	2.1046E+14	1.1850E+02
C16 ₃₀	1.00	9.3333E-01	0.0	0.20	8.6371E-01	2.2986E-03
C17 ₁₀	1.00	1.4457E+01	0.0	0.00	N/A	1.4995E-02
C17 ₃₀	1.00	9.2735E+02	0.0	0.00	N/A	6.2842E-02
C18 ₁₀	1.00	7.8021E-01	0.0	0.20	1.3353E+03	2.0852E-03
C18 ₃₀	1.00	2.1776E+02	0.0	0.00	N/A	3.5281E-03

TABLE S.VII
STATISTICAL RESULTS FOR THE COST-EFFECTIVENESS ANALYSIS OF GRM

Inst.	SR	AT	Inst.	SR	AT	Inst.	SR	AT
G01	1.00	0.00	G02	1.00	0.00	G03	1.00	0.00
G04	1.00	0.00	G05	1.00	0.00	G06	1.00	0.55
G07	1.00	0.00	G08	1.00	0.00	G09	1.00	0.00
G10	1.00	0.00	G11	1.00	0.00	G12	1.00	0.00
G13	1.00	0.00	G14	1.00	0.00	G15	1.00	0.00
G16	1.00	0.00	G17	1.00	0.00	G18	1.00	0.00
G19	1.00	0.00	G21	1.00	2.10	G23	1.00	0.00
G24	1.00	0.00	C01 ₁₀	1.00	0.00	C02 ₁₀	1.00	1.25
C03 ₁₀	1.00	1.15	C05 ₁₀	0.85	1.80	C06 ₁₀	1.00	0.80
C07 ₁₀	1.00	0.00	C08 ₁₀	1.00	0.00	C09 ₁₀	1.00	1.35
C10 ₁₀	1.00	1.40	C13 ₁₀	1.00	0.00	C14 ₁₀	1.00	0.00
C15 ₁₀	1.00	0.85	C16 ₁₀	1.00	0.85	C17 ₁₀	1.00	1.40
C18 ₁₀	1.00	1.75	C01 ₃₀	1.00	0.00	C02 ₃₀	1.00	1.00
C05 ₃₀	1.00	2.00	C06 ₃₀	1.00	2.00	C07 ₃₀	1.00	0.00
C08 ₃₀	1.00	0.00	C09 ₃₀	1.00	1.00	C10 ₃₀	1.00	1.00
C13 ₃₀	1.00	0.00	C14 ₃₀	1.00	0.00	C15 ₃₀	1.00	1.00
C16 ₃₀	1.00	3.10	C17 ₃₀	1.00	1.00	C18 ₃₀	1.00	1.00

TABLE S.VIII
 RUN-TIME RESULTS OF THE SIX COMPARED ALGORITHMS AND SADE-SA-GRM

Inst. & Time Unit (min)	GLoSADE	DSI-C ² oDE	MPMLS	SA-TSDE	SParEA	eToSA-DE	SaDE-SA-GRM
C01 ₁₀	0.99	0.85	0.10	0.12	6.6	7.8	4.57
C02 ₁₀	0.91	0.40	0.11	0.03	12.6	24	5.30
C03 ₁₀	0.90	1.08	0.11	0.03	60	42	5.07
C04 ₁₀	0.91	0.62	0.11	0.03	37.8	18	4.98
C05 ₁₀	0.90	0.35	0.10	0.03	76.8	18.6	5.12
C06 ₁₀	0.90	0.43	0.10	0.03	87	20.4	5.46
C07 ₁₀	0.90	0.68	0.11	0.08	138.6	7.8	5.36
C08 ₁₀	0.91	0.59	0.11	0.14	150	8.4	5.29
C09 ₁₀	0.92	0.21	0.11	0.03	180.6	42	5.09
C10 ₁₀	0.94	0.20	0.11	0.03	208.2	25.2	5.05
C11 ₁₀	0.95	0.94	0.11	0.03	233.4	25.8	4.51
C12 ₁₀	0.95	1.05	0.11	0.03	174	19.2	5.06
C13 ₁₀	0.91	0.43	0.11	0.07	142.2	10.2	6.68
C14 ₁₀	0.92	0.44	0.11	0.10	150.6	11.4	5.11
C15 ₁₀	0.92	0.46	0.10	0.07	159.6	21	4.96
C16 ₁₀	0.93	0.45	0.10	0.05	137.4	34.8	5.02
C17 ₁₀	0.91	0.29	0.10	0.05	183	20.4	5.28
C18 ₁₀	0.88	0.68	0.10	0.06	254.4	21.6	4.56

APPENDIX A
EXAMPLE OF GRM EMPLOYMENT

Considering the test function C16₁₀ from CEC2010 [1], the optimization problem is formulated as follows:

$$\begin{aligned}
 \text{minimize} \quad & f(\mathbf{x}) = \sum_{i=1}^{10} \frac{z_i^2}{4000} - \prod_{i=1}^{10} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1, & \mathbf{x} = (x_1, \dots, x_{10}), \quad \mathbf{z} = \mathbf{x} - \mathbf{o} \\
 \text{subject to} \quad & g_1(\mathbf{x}) = \sum_{i=1}^{10} [z_i^2 - 100 \cos(\pi z_i) + 10] \leq 0 \\
 & g_2(\mathbf{x}) = \prod_{i=1}^{10} z_i \leq 0 \\
 & h_1(\mathbf{x}) = \sum_{i=1}^{10} (z_i \sin(\sqrt{|z_i|})) = 0, \\
 & -10 \leq x_i \leq 10, & i = 1, \dots, 10, \\
 & \mathbf{o} = (0.365972807627352, 0.429881383400138, \\
 & \quad -0.420917679577772, 0.984265986788929, \\
 & \quad 0.324792771198785, 0.463737106835568, \\
 & \quad 0.989554882052943, 0.307453878359996, \\
 & \quad 0.625094764380575, -0.358589007202526)
 \end{aligned} \tag{1}$$

The objective function f , inequality constraints g_1 and g_2 , and equality constraint h_1 are shifted by $\mathbf{o}$. As these functions are provided in a black-box format, gradients are typically approximated numerically. For instance, the gradient of $h_1(\mathbf{x})$ with respect to the i th variable is approximated as:

$$\frac{\partial h_1}{\partial x_i}(\mathbf{x}) = \frac{h_1(x_1, \dots, x_i + \delta, \dots, x_n) - h_1(\mathbf{x})}{\delta}, \tag{2}$$

where $\delta = 0.0001$ in this study.

Based on the experimental record from the median of 20 independent runs, the initial infeasible solution and the corresponding number of function evaluations ($nFEs$) input to GRM are $\mathbf{x} = (-0.4357, 2.6032, -6.2176, 7.0225, 3.3472, -1.5740, -2.0282, 2.2224, 8.6937, -6.0925)$ and 347, respectively. To approximate gradients of constraints g_1 , g_2 , and h_1 for each variable using (2), ten intermediate solutions are generated as follows:

$$\begin{aligned}
 \mathbf{x}_1 &= (x_1 + \delta) + x_2, \dots, x_{10}, \\
 \mathbf{x}_2 &= x_1 + (x_2 + \delta), \dots, x_{10}, \\
 &\quad \vdots \\
 \mathbf{x}_{10} &= x_1 + x_2, \dots, (x_{10} + \delta).
 \end{aligned} \tag{3}$$

Evaluating these intermediate solutions incurs ten additional expensive function evaluations, thus increasing $nFEs$ from 347 to 357. After evaluating these intermediate points, the gradient matrix $\nabla_{\mathbf{x}}G$ is constructed as:

$$\nabla_{\mathbf{x}}G = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(\mathbf{x}), & \frac{\partial g_1}{\partial x_2}(\mathbf{x}), & \dots, & \frac{\partial g_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial g_2}{\partial x_1}(\mathbf{x}), & \frac{\partial g_2}{\partial x_2}(\mathbf{x}), & \dots, & \frac{\partial g_2}{\partial x_n}(\mathbf{x}) \\ \frac{\partial h_1}{\partial x_1}(\mathbf{x}), & \frac{\partial h_1}{\partial x_2}(\mathbf{x}), & \dots, & \frac{\partial h_1}{\partial x_n}(\mathbf{x}) \end{bmatrix}. \tag{4}$$

In our illustrative example, the numerical gradient matrix is:

$$\nabla_{\mathbf{x}}G = \begin{bmatrix} 0, 0, -1.06032245 \\ 0, 0, -1.0664004 \\ 0, 0, 0.22402536 \\ 0, 0, 0.31988515 \\ 0, 0, -0.84085059 \\ 0, 0, -1.09168769 \\ 0, 0, -0.84235939 \\ 0, 0, -1.11118591 \\ 0, 0, 1.05986371 \\ 0, 0, 0.19898131 \end{bmatrix}^T. \tag{5}$$

Subsequently, the Moore-Penrose pseudo-inverse is employed to compute:

$$-\nabla_{\mathbf{x}}^{-1}G = \begin{bmatrix} 0, 0, -0.14290008 \\ 0, 0, -0.1437192 \\ 0, 0, 0.03019199 \\ 0, 0, 0.04311105 \\ 0, 0, -0.11332177 \\ 0, 0, -0.14712718 \\ 0, 0, -0.11352511 \\ 0, 0, -0.14975496 \\ 0, 0, 0.14283825 \\ 0, 0, 0.02681679 \end{bmatrix}. \quad (6)$$

Since $\mathbf{x}$ has already been evaluated, the constraint violation vector $C(\mathbf{x})$ is directly available. In this example, $C(\mathbf{x}) = [0, 0, 0.1627]$. A new candidate solution $\mathbf{x}'$ is then generated as:

$$\mathbf{x}' = -\nabla_{\mathbf{x}}^{-1}GC(\mathbf{x}) + \mathbf{x} \quad (7)$$

yielding:

$$\mathbf{x}' = (-0.4124, 2.6266, -6.2225, 7.0154, 3.3656, -1.5500, -2.0097, 2.2467, 8.6704, -6.0969). \quad (8)$$

Evaluating $\mathbf{x}'$ requires one additional expensive function evaluation, thus incrementing *nFEs* from 357 to 358. Correspondingly, the constraint violation vector at $\mathbf{x}'$ is computed as $C(\mathbf{x}') = [0, 0, 0.0002]$.

It is evident that a clear improvement exists from solution $\mathbf{x}$ to $\mathbf{x}'$. Consequently, GRM is not terminated but proceeds to utilize $\mathbf{x}'$ for the subsequent iteration. Different from the initial solution $\mathbf{x}$, the gradient matrix of constraints g_1 , g_2 , and h_1 at $\mathbf{x}'$ is not re-approximated; instead, the previously $\nabla_{\mathbf{x}}G$ is reused. Specifically, the next candidate solution is computed as:

$$\mathbf{x}'' = -\nabla_{\mathbf{x}}^{-1}GC(\mathbf{x}') + \mathbf{x}', \quad (9)$$

yielding:

$$\mathbf{x}'' = (-0.4123, 2.6266, -6.2225, 7.0154, 3.3656, -1.5500, -2.0097, 2.2467, 8.6704, -6.0969). \quad (10)$$

Evaluating $\mathbf{x}''$ necessitates one additional expensive FE, thereby increasing *nFEs* from 358 to 359. The constraint violation vector for $\mathbf{x}''$ is obtained as $C(\mathbf{x}'') = [0, 0, 6.5294E-07]$. Similarly, in the third iteration, the solution is updated to

$$\mathbf{x}''' = -\nabla_{\mathbf{x}}^{-1}GC(\mathbf{x}'') + \mathbf{x}'', \quad (11)$$

with constraint violation vector $C(\mathbf{x}''') = [0, 0, 1.8587E-09]$. In this illustrative example, six iterations are executed, ultimately achieving zero overall constraint violations.

Fig. S.3 schematically illustrates the GRM procedure within a two-dimensional representation. According to $\mathbf{x}$ and $C(\mathbf{x})$, although the input solution $\mathbf{x}$ for GRM is infeasible, it satisfies the inequalities g_1 and g_2 . The activation of GRM indicates that neither global nor local surrogate-driven evolution can further enhance the solution quality, specifically highlighting the inability of these methods alone to handle the equality constraint h_1 . The initial infeasible solution (represented by the black point $\mathbf{x}$) and the iteratively improved solutions (depicted as gray points $\mathbf{x}'$, $\mathbf{x}''$, $\mathbf{x}'''$) progressively approach feasibility along the gradient direction indicated by the red line. The gradient information, numerically approximated only once, effectively addresses the equality constraint h_1 . This demonstrates GRM's effectiveness in resolving constraints that surrogate-driven evolution methods alone cannot adequately manage.

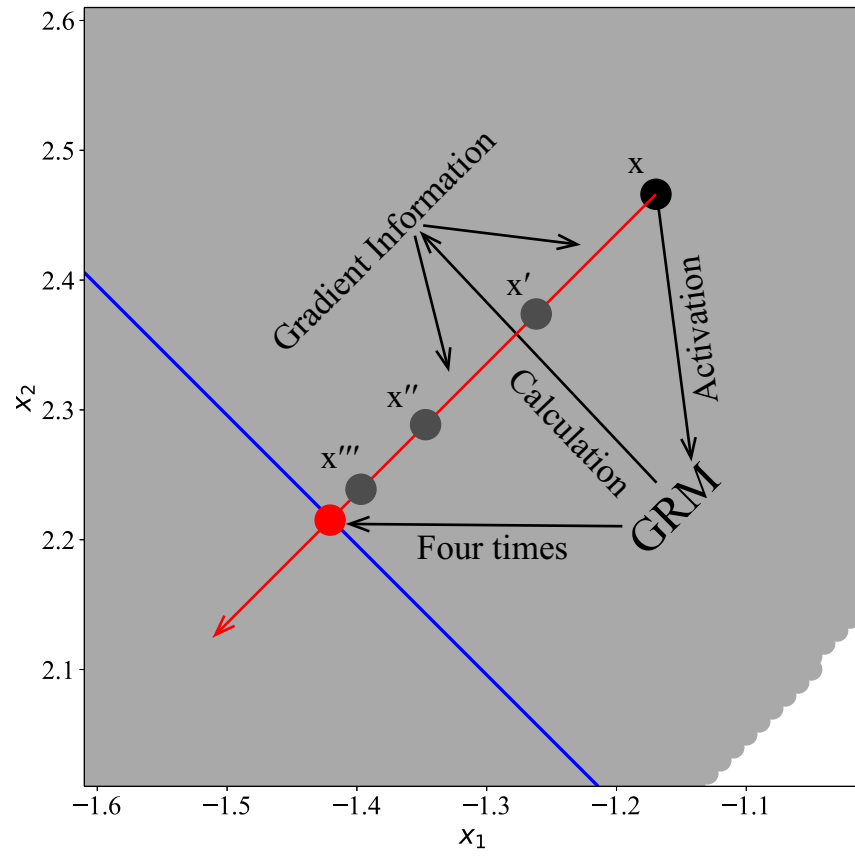


Fig. S.3. A schematic illustration of the GRM process applied to the given example. The black point x represents the initial infeasible solution. The gray-shaded region comprises solutions satisfying constraints g_1 and g_2 but violating the equality constraint h_1 , whereas the blue line denotes solutions satisfying h_1 but violating g_1 and g_2 . The red line indicates the gradient direction of the constraint h_1 at the solution x .

REFERENCES

- [1] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," *Nanyang Technological University, Singapore*, vol. 24, p. 910, 2010.