Global and Local Surrogate-Assisted Differential Evolution for Expensive Constrained Optimization with Inequality Constraints

Yong Wang, *Senior Member*, *IEEE*, Da-Qing Yin, Shengxiang Yang, *Senior Member*, *IEEE*, and Guangyong Sun

***Abstract*—For expensive constrained optimization problems, the computation of objective function and constraints is very time-consuming. This paper proposes a novel global and local surrogate-assisted differential evolution for solving expensive constrained optimization problems with inequality constraints. The proposed method consists of two main phases: global surrogate-assisted phase and local surrogate-assisted phase. In the global surrogate-assisted phase, differential evolution serves as the search engine to produce multiple trial vectors. Afterward, the generalized regression neural network is used to evaluate these trial vectors. In order to select the best candidate from these trial vectors, two rules are combined. The first is the feasibility rule, which at first guides the population toward the feasible region, and then toward the optimal solution. In addition, the second rule puts more emphasis on the solution with higher predicted uncertainty, and thus alleviates the inaccuracy of the surrogates. In the local surrogate-assisted phase, the interior point method coupled with radial basis function is utilized to refine each individual in the population. During the evolution, the global surrogate-assisted phase has the capability to promptly locate the promising region and the local surrogate-assisted phase is able to speed up the convergence. Therefore, by combining these two important elements, the number of fitness evaluations can be reduced remarkably. The proposed method has been tested on numerous benchmark test functions from three test suites and two real-world cases. The experimental results demonstrate that the performance of the proposed method is better than that of other state-of-the-art methods.**

***Index Terms*—Expensive constrained optimization problems, surrogate model, global search, local search, differential evolution**

This work was supported in part by the Innovation-Driven Plan in Central South University under Grant 2018CX010, in part by the National Natural Science Foundation of China under Grants 61673397 and 61673331, in part by the EU Horizon 2020 Marie Sklodowska-Curie Individual Fellowships (Project ID: 661327), in part by the Engineering and Physical Sciences Research Council of UK under Grant EP/K001310/1, and in part by the Hunan Provincial Natural Science Fund for Distinguished Young Scholars (Grant No. 2016JJ1018).

Y. Wang is with the School of Information Science and Engineering, Central South University, Changsha 410083, China, and also with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK. (Email: ywang@csu.edu.cn).

D.-Q. Yin is with the School of Information Science and Engineering, Central South University, Changsha 410083, China. (Email: yindaqing@csu.edu.cn)

S. Yang is with the Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK, and also with the College of Information Engineering, Xiangtan University, Xiangtan 411105, China. (Email: syang@dmu.ac.uk)

G. Sun is with the School of Aerospace, Mechanical and Mechatronic Engineering, Faculty of Engineering, University of Sydney, Sydney, NSW 2006, Australia, and also with the State Key Laboratory of Advanced Design and Manufacture for Vehicle Body, Hunan University, Changsha 410082, China (e-mail: guangyong.sun@sydney.edu.au).

I. Introduction

In many science and engineering areas such as computational fluid dynamics [1], computational structural mechanics [2], and computational electromagnetics [3], optimization problems always include one objective function and various constraints. Moreover, for some of them, the computation of objective function and constraints is extremely expensive (for example, one simulation of typical computational electromagnetics may take 20 minutes [4]). This kind of optimization problems is considered to be expensive constrained optimization problems (ECOPs). An ECOP with inequality constraints can be formulated as follows:

$$\begin{aligned} &\text{minimize: } f(\vec{x}) \\ &\text{Subject to: } g_j(\vec{x}) \le 0,\ j = 1,\ldots,p \end{aligned} \tag{1}$$

where $\vec{x} = (x_1,\ldots,x_D) \in S$ is the decision vector, $L_i \le x_i \le U_i$ $(i \in \{1,\ldots,D\})$ is the ith decision variable, D is the number of decision variables, L_i and U_i are the lower and upper bounds of x_i respectively, $S = \prod_{i=1}^{D}[L_i, U_i]$ is the search space or decision space, $f(\vec{x})$ is the objective function, $g_j(\vec{x})$ is the jth inequality constraint, and p is the number of inequality constraints. Due to the existence of constraints, when evaluating a decision vector $\vec{x}$, we need to consider its degree of constraint violation. The degree of constraint violation of $\vec{x}$ on the jth constraint can be computed as follows:

$$G_j(\vec{x}) = \max(0, g_j(\vec{x})),\ j = 1,\ldots,p \tag{2}$$

Afterward, $G(\vec{x}) = \sum_{j=1}^{p} G_j(\vec{x})$ denotes the degree of constraint violation of $\vec{x}$ on all constraints.

Evolutionary algorithms (EAs) are a class of population-based heuristic search approaches, which have been successfully applied to solve different types of optimization problems. EAs usually implement crossover, mutation, and selection operators to improve the quality of each individual in the population generation by generation. In general, EAs need a large number of fitness evaluations to obtain the global optimum of an optimization problem.

In the evolutionary computation research community, how to deal with expensive unconstrained optimization problems (EUOPs) has attracted much attention. Owing to the computationally expensive nature of fitness evaluations, applying EAs to solve EUOPs in a straightforward way is highly time-consuming and even impractical. Over the past

fifteen years, surrogate-assisted EAs (SAEAs) have been widely accepted as one of the most popular methods for solving EUOPs. The basic idea behind SAEAs is to build a surrogate model to approximate the original expensive objective function. Since the surrogate model is much more computationally efficient than the original objective function, the computational cost can be reduced significantly. At present, a variety of surrogate models has been developed, such as polynomial regression, Gaussian process (GP), radial basis function (RBF), artificial neural network, and support vector machine (SVM). In order to find the global optimum within a limited budget of fitness evaluations, SAEAs should combine the surrogate model with the original objective function in an effective way. There are three kinds of methods, namely evolution control, surrogate-assisted prescreening method, and surrogate-assisted local search, to address the above issue. Evolution control, proposed by Jin *et al.* [5], is a simple yet popular framework for managing surrogates, which includes individual-based control and generation-based control. In individual-based control, some of individuals are evaluated with the original objective function and the remaining individuals are evaluated with surrogates at each generation. In contrast, with respect to generation-based control, all individuals in the population are evaluated with surrogates in some generations, while in the rest of generations the original objective function is used for fitness evaluations of all individuals. For surrogate-assisted pre-screening methods [6], [7], surrogates are used to preselect a subset of promising individuals among a number of trial offspring. Afterward, the chosen individuals are re-evaluated with the original objective function. Regarding surrogate-assisted local search [8], [9], the surrogate of objective function and its gradient information are utilized to generate an offspring in each iteration. Finally, the resulting offspring is re-evaluated with the original objective function. In 2011, Jin [10] carried out a comprehensive survey on SAEAs.

On the other hand, recent two decades have witnessed the significant development of constraint-handling techniques for inexpensive constrained optimization problems. The current popular constraint-handling techniques can be classified into three categories [11], [12], [13]: methods based on penalty functions [14]-[16], methods based on the preference of feasible solutions over infeasible solutions [17]-[20], and methods based on multiobjective optimization [21]-[25]. For the methods based on penalty functions, an infeasible solution is penalized according to its degree of constraint violation. With respect to the methods based on the preference of feasible solutions over infeasible solutions, either objective function or the degree of constraint violation is used to compare the individuals in the population. Moreover, in this kind of methods, the search is biased toward the feasible region. In terms of the methods based on multiobjective optimization, the constraints are treated as one or more objective functions. The interested reader is referred to a recent survey paper on constraint-handling techniques [26].

Although EUOPs and constraint-handling techniques have been actively investigated, not much work has been done to solve ECOPs in the evolutionary computation research community. When solving ECOPs by EAs, most of the current methods adopt global surrogates or local surrogates individually; therefore, their effectiveness and efficiency are limited. Additionally, how to improve the quality of surrogates during the evolution and how to combine surrogates with constraint-handling techniques have not been studied in depth. Motivated by the above considerations, this paper proposes a global and local surrogate-assisted differential evolution (DE) called GLoSADE for solving ECOPs with inequality constraints. To the best of our knowledge, GLoSADE is the first attempt to combine both global and local surrogates in expensive constrained evolutionary optimization. GLoSADE consists of two important phases: global surrogate-assisted phase and local surrogate-assisted phase.

The main contributions of this paper can be summarized as follows.

- The global surrogate-assisted phase makes use of DE to generate multiple trial vectors and the generalized regression neural network (GRNN) to evaluate them. It is worth noting that the training procedure of GRNN is one-passing learning, and the computational cost is thus very low. Subsequently, two rules are combined to select the most potential candidate from the trial vectors. The first rule, named the feasibility rule [17], selects the best infeasible solution if the population only contains infeasible solutions or the best feasible solution if the population contains at least one feasible solution, with the aim of at first guiding the population toward the feasible region, and then toward the optimal solution. The second rule, called the uncertainty-based rule, chooses the individual with the highest predicted uncertainty, which can improve the quality of surrogates.
- In the local surrogate-assisted phase, the interior point method refines each individual in the population in an iterative manner. In each iteration, RBF is used to construct a gradient descent direction and a step size to create a new solution.
- Because of the powerful global search ability of DE and the good generalization performance of GRNN, the global surrogate-assisted phase is capable of locating the promising region promptly. On the other hand, the local surrogate-assisted phase is able to speed up the convergence. Hence, GLoSADE attempts to strike a balance between the effectiveness and efficiency.
- Systematic experiments have been conducted to study the performance of GLoSADE and compare it with three other state-of-the-art methods on the benchmark test functions at IEEE CEC2006 [27], IEEE CEC2010 [28], and IEEE CEC2017 [29]. Moreover, GLoSADE has been successfully applied to solve two practical ECOPs, i.e., the optimal shape design of transonic airfoil and the topology optimization of energy-absorbing component.

The rest of this paper is organized as follows. Section II briefly introduces DE and surrogates. Section III describes the related work. The proposed method, GLoSADE, is elaborated in Section IV. Section V and Section VI present the experimental studies on benchmark test functions from three test suites and two real-world applications, respectively. Finally, Section VII concludes this paper.

II. Differential Evolution And Surrogates

A. Differential Evolution (DE)

DE, proposed by Storn and Price [30], is a very popular EA paradigm for solving complex optimization problems. In this paper, DE is employed as the search engine. Similar to other EAs, DE is also a population-based optimizer [31], [32]. Suppose that $P=\{\vec{x}_1,\ldots,\vec{x}_{NP}\}$ denotes the population and $\vec{x}_k=(x_{k,1},\ldots,\ x_{k,D})$ is the *k*th individual (also called the *k*th target vector) in *P*. Over the course of evolution, DE consists of three main operators, i.e., mutation, crossover and selection.

Firstly, the mutant vector $\vec{v}_k=(v_{k,1},\ldots,v_{k,D})$ is generated by a mutation operator for each target vector $\vec{x}_k$. The commonly used mutation operators are listed as follows:

- DE/rand/1:

$$\vec{v}_k=\vec{x}_{r_1}+F\times(\vec{x}_{r_2}-\vec{x}_{r_3}) \tag{3}$$

- DE/best/1:

$$\vec{v}_k=\vec{x}_{best}+F\times(\vec{x}_{r_1}-\vec{x}_{r_2}) \tag{4}$$

- DE/current-to-rand/1:

$$\vec{v}_k=\vec{x}_k+F\times(\vec{x}_{r_1}-\vec{x}_k)+F\times(\vec{x}_{r_2}-\vec{x}_{r_3}) \tag{5}$$

- DE/current-to-best/1:

$$\vec{v}_k=\vec{x}_k+F\times(\vec{x}_{best}-\vec{x}_k)+F\times(\vec{x}_{r_1}-\vec{x}_{r_2}) \tag{6}$$

where r_1, r_2, and r_3 are three mutually different integers uniformly generated from $\{1,\ldots,k-1,k+1,\ldots,D\}$, $\vec{x}_{best}$ is the best individual in the population, and *F* is the scaling factor.

Afterward, the crossover operator is implemented on $\vec{x}_k$ and $\vec{v}_k$ to generate the trial vector $\vec{u}_k=(u_{k,1},\ldots,u_{k,D})$. The binomial crossover is expressed as follows:

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

where $rand_j$ is a random number on the interval $[0,1]$, j_{rand} is a integer randomly selected form $\{1,\ldots,D\}$, and $CR\in[0,1]$ is the crossover control parameter.

Finally, the selection operator chooses the better one between $\vec{x}_k$ and $\vec{u}_k$ into the next generation.

B. Surrogates

In this paper, GRNN and RBF are used to construct the global and local surrogates, respectively. Next, we give a brief introduction to them.

- *Generalized Regression Neural Network (GRNN)*

GRNN, proposed by Specht [33], is a memory-based neural network, which can approximate any underlying regression surface in theory. Unlike other neural networks, such as back-propagation neural network, GRNN is a one-pass learning algorithm. Thus, the time-saving GRNN is very suitable for dealing with expensive optimization problems. Assuming that $f(\vec{x},y)$ denotes the joint probability function of a random vector $\vec{x}$ and a random variable *y*. The conditional mean of y on $\vec{x}$ is given by

$$E[y\mid\vec{x}]=\frac{\int_{-\infty}^{\infty}yf(\vec{x},y)dy}{\int_{-\infty}^{\infty}f(\vec{x},y)dy} \tag{8}$$

Given the data points $\{(\vec{x}_i,y_i)\mid\vec{x}_i\in\Re^D,i=1,\ldots,N\}$, the joint probability density function can be estimated by Parzen nonparametric estimator [34] as follows:

$$\hat{f}(\vec{x},y)=\frac{1}{N(2\pi)^{(D+1)/2}\sigma^{D+1}}\times \sum_{i=1}^{N}\exp\left[-\frac{(\vec{x}-\vec{x}_i)(\vec{x}-\vec{x}_i)^T}{2\sigma^2}\right]\exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] \tag{9}$$

where σ is a user-defined smoothness parameter.

By combining Eq. (8) with Eq. (9), the conditional mean of y can be calculated as

$$E[y\mid\vec{x}]=\frac{\sum_{i=1}^{N}y_i\exp\left[-\frac{(\vec{x}-\vec{x}_i)(\vec{x}-\vec{x}_i)^T}{2\sigma^2}\right]}{\sum_{i=1}^{N}\exp\left[-\frac{(\vec{x}-\vec{x}_i)(\vec{x}-\vec{x}_i)^T}{2\sigma^2}\right]} \tag{10}$$

As a result, when point $\vec{x}$ is given, the predicted output on $\vec{x}$ is equal to $E[y\mid\vec{x}]$.

- *Radial Basis Function (RBF)*

RBF is a kind of interpolation model and has been widely applied to various scientific and engineering fields. RBF uses a weighted sum of simple basis functions to interpolate the scatter points. Given the data points $\{(\vec{x}_i,y_i)\mid\vec{x}_i\in\Re^D,\ i=1,\ldots,N\}$, RBF approximates the underlying function as follows:

$$\hat{f}(\vec{x})=\sum_{i=1}^{N}\omega_i\phi(\vec{x}-\vec{x}_i) \tag{11}$$

where ω_i is the weight coefficient and $\phi(\cdot)$ is a basis function. Many forms of the basis function can be used here. The cubic form $\phi(r)=r^3$ is chosen in this paper. The previous studies have revealed that the cubic form outperforms the other kinds of basis function [35]. In addition, the weight vector $\vec{\omega}=(\omega_i,\ldots,\omega_N)^T$ can be computed as follows:

$$\vec{\omega}=(\Phi^T\Phi)^{-1}\Phi^T\vec{y} \tag{12}$$

where $\vec{y}=(y_1,\ldots,y_N)^T$ is the output vector and Φ is the following matrix:

$$\Phi=\begin{bmatrix}\phi(\vec{x}_1-\vec{x}_1) & \cdots & \phi(\vec{x}_1-\vec{x}_N)\\ \vdots & \ddots & \vdots\\ \phi(\vec{x}_N-\vec{x}_1) & \cdots & \phi(\vec{x}_N-\vec{x}_N)\end{bmatrix} \tag{13}$$

III. The Related Work

Solving ECOPs by EAs is one of the most important challenges in the evolutionary computation research community. This section presents a brief review of EAs for ECOPs from two aspects: surrogate-assisted prescreening methods and surrogate-assisted local search.

The primary idea of the surrogate-assisted prescreening methods is the following. Firstly, a number of trial offspring are generated by EA operators. Secondly, surrogates are utilized to

evaluate the trial offspring and a selection strategy is presented to preselect some desired candidates from the trial offspring. Finally, these candidates are re-evaluated with the original objective function and constraints. For example, Krempser *et al.* [36] proposed a DE assisted by surrogates for structural optimization problems. In this method, four DE variants are adopted to generate four trial offspring for each individual in one generation. Afterward, the similarity-based surrogate is applied to evaluate the trial offspring to reduce the number of fitness evaluations. Mezura-Montes *et al.* [37] developed a fitness inheritance-based evolution strategy for constrained optimization problems. To perform the fitness inheritance well, they also introduced two important parameters, i.e., inheritance ratio which determines the percentage of offspring to use the fitness inheritance mechanism, and replacement ratio which determines the percentage of offspring with inherited fitness values to survive into the next generation. Moreover, the feasibility rule is chosen as the constraint-handling technique. Runarsson [38] constructed two surrogates, one for the objective function and the other for the penalty function. After the trial offspring are generated by (μ,λ)-evolution strategy, stochastic ranking [19] is employed to select the best individual from the trial offspring. Büche *et al.* [39] proposed a novel method based on Gaussian process. In this method, Gaussian process is constructed only in the neighborhood of the current best individual. To balance the exploration and exploitation, a merit function proposed by Torczon and Trosset [40] is used to choose the best candidate among the trial offspring. In [41], Emmerich *et al.* proposed an efficient search method based on EA assisted by local Gaussian random field metamodels. In this approach, a merit function proposed by Schonlau *et al.* [42] is used to rank the trial offspring and then prescreen the promising trial offspring. Regies [43] developed a RBF-assisted evolutionary programming for high-dimensional constrained expensive black-box optimization problems. This method makes use of the mutation operator of evolutionary programming to generate a very large number of trial offspring for each parent and RBF to evaluate them. Subsequently, the feasibility rule selects the best candidate among the trial offspring. Based on the work in [44], Bagheri *et al.* [45] self-adjusted the parameters in RBF approximation for ECOPs.

The surrogate-assisted local search improves the quality of each individual in the population in an iterative manner. During the evolution, the local search exploits the information provided by surrogates of the original objective function and constraints to produce a sequence of offspring. After that, the original objective function and constraints are used to re-evaluate the final offspring. For instance, Ong *et al.* [46] proposed a surrogate-assisted parallel evolutionary optimization algorithm to solve ECOPs. In this method, the local search, named feasible sequential quadratic programming [47], is integrated with EA in the spirit of Lamarckian learning. Moreover, a trust-region approach is employed in the local search for interleaving use of the original objective function and constraints with the RBF surrogates. Goh *et al.* [48] proposed a surrogate-assisted memetic co-evolutionary framework to solve ECOPs, where the population is divided into several subpopulations with random decomposition. After the offspring are generated in the co-evolutionary optimization phase, the surrogate-assisted local search is applied to each offspring. Moreover, a multiobjective ranking scheme is designed to handle constraints. Handoko *et al.* [49] proposed a SVM-assisted memetic algorithm for the constrained optimization problems with single equality constraint. In this method, SVM is used to predict the relative position of individuals with respect to the boundary of the feasible region, and then the local search is only applied to the individuals located near the boundary. In [50], Handoko *et al.* further developed an effective chaperone for constrained memetic algorithm, called feasibility structure modeling. In this approach, SVM identifies the location of an individual in the search space as follows: deep inside the infeasible region, near the feasibility boundary, or deep inside the feasible region. Subsequently, the location information is utilized to determine whether or not the local search should be applied to an individual. In [51], Regis designed a trust region-like approach to improve the best solution at the end of each generation.

From the above introduction, it is clear that the surrogate-assisted prescreening methods usually employ global surrogates. Although global surrogates are able to enhance the global search ability by exploring the whole search space, the accuracy of the obtained solutions may be low. In addition, the surrogate-assisted local search usually takes advantage of local surrogates. Local surrogates can speed up the convergence by probing a relatively small search region; however, they may ignore some promising regions in the search space. Therefore, the capabilities of the current methods to converge to the global optimum and keep a fast convergence speed simultaneously are limited, especially when the fitness landscape of the search space or the structure of the feasible region is complicated.

IV. Proposed Approach

A. Motivation

As discussed in Section III, the current methods either employ global surrogates to explore the search space or utilize local surrogates to accelerate the convergence. An interesting question is whether we can integrate the advantages of both global surrogates and local surrogates. We have observed, however, that the above issue has not yet been systematically investigated in the community of expensive constrained evolutionary optimization. In addition, to prevent the population from being misled by a false optimum introduced by surrogates, it is necessary to refine the quality of surrogates gradually. In principle, when solving ECOPs by EAs, constraint-handling technique and surrogates are two critical components. Therefore, the performance of a method will depend strongly on the way of how to integrate them.

Based on the above considerations, we propose a global and local surrogate-assisted DE which is the first attempt to synthesize global and local surrogates to solve ECOPs. GLoSADE consists of two important phases at each generation: global surrogate-assisted phase and local surrogate-assisted phase. In the former, on the one hand, DE/rand/1/bin generates many trial vectors for each individual with a probability 0.5. Then GRNN is used to evaluate these trial vectors and the best

trial vector is selected based on the feasibility rule. The aim of the above process is to locate the promising region quickly. On the other hand, DE/current-to-rand/1 is another choice to produce multiple trial vectors for each individual with a probability 0.5. Afterward, the uncertainties of these trial vectors are calculated and the trial vector with the highest uncertainty is chosen, with the purpose of improving the quality of surrogates. Additionally, in the latter, the interior point method and RBF are performed as the local search on each individual in the population, thus enhancing the accuracy of solution. It is necessary to emphasize that GRNN and RBF are both multi-output models. A multi-output model can approximate multiple functions at the same time. In this paper, we separately approximate the original objective function and each constraint by surrogates throughout the evolutionary process.[1] On the whole, GLoSADE also provides an effective combination of surrogates and constraint-handling techniques.

B. GLoSADE

The framework of GLoSADE is given in Fig. 1. Firstly, the initial population P_0 is produced by Latin hypercube design [52] and the initial archive *A* is constructed. At each generation, both the global surrogate-assisted phase and the local surrogate-assisted phase are performed for each individual in the population. In the global surrogate-assisted phase, GRNN is constructed as the global surrogate to approximate the original objective function and constraints. Afterward, an offspring $\vec{g}_{k,t}$ is generated for each individual $\vec{x}_{k,t}$ in P_t through the global surrogate-assisted search as explained in Section IV-C. $\vec{g}_{k,t}$ is re-evaluated with the original objective function and constraints and its information is stored into *A*. Subsequently, $\vec{g}_{k,t}$ is compared with $\vec{x}_{k,t}$ based on the feasibility rule and the better one survives into Q_t. In the local surrogate-assisted phase, an offspring $\vec{l}_{k,t}$ is generated for each individual $\vec{x}_{k,t}$ in Q_t via the local surrogate-assisted search as explained in Section IV-D. Then, $\vec{l}_{k,t}$ is re-evaluated with the original objective function and constraints and its information is also stored into *A*. Subsequently, the better one between $\vec{l}_{k,t}$ and $\vec{x}_{k,t}$ is selected based on the feasibility rule and put into P_{t+1}. The above procedure is repeated until the maximum number of fitness evaluations (FEs) is reached.

Surrogates play a crucial role in solving ECOPs by EAs. In [53], Lim *et al.* pointed out that SAEAs work on two major tasks: 1) alleviating the "curse of uncertainty", and 2) benefiting from the "blessing of the uncertainty". The "curse of uncertainty" can be briefly defined as the phenomenon where SAEAs search to stall or converge to a false optimum due to the approximation error of a surrogate. The "blessing of uncertainty" means that the approximation error of a surrogate can be helpful to smooth the rough fitness landscape and promptly locate the promising region. It is clear that both the "curse of uncertainty" and the "blessing of the uncertainty" are related to the approximation error of a surrogate, which represent the shortcoming and advantage of the approximation error of a surrogate, respectively. To alleviate the "curse of uncertainty", it is expected to build an accurate surrogate to approximate the fitness landscape in the specified search region, thus reducing the approximation error. Under this condition, the surrogate is always computationally expensive. However, the "blessing of uncertainty" implies that we can use an inaccurate and cheap surrogate to benefit from the approximation error. Obviously, the above two aspects lead to a contradiction. To address this contradiction, in this paper we employ cheap GRNN to construct a global surrogate and

1. $t = 1$; // t denotes the generation number
2. Generate the initial population $P_t = \{\vec{x}_{1,t}, \ldots, \vec{x}_{NP,t}\}$ by Latin hypercube design;
3. Calculate the values of the original objective function and constraints of P_t: $\{(f(\vec{x}_{k,t}), g_1(\vec{x}_{k,t}), \ldots, g_p(\vec{x}_{k,t})) \mid k = 1, \ldots, NP\}$;
4. Initialize the archive: $A = \{(\vec{x}_{k,t}, f(\vec{x}_{k,t}), g_1(\vec{x}_{k,t}), \ldots, g_p(\vec{x}_{k,t})) \mid k = 1, \ldots, NP\}$;
5. $FEs = NP$; // FEs denotes the number of fitness evaluations
6. **While** $FEs < MaxFEs$
7. $Q_t = \varnothing$ and $P_{t+1} = \varnothing$;
   // The global surrogate-assisted phase
8. Construct GRNN by utilizing all the individuals in A to approximate the original objective function and constraints;
9. **For** each individual $\vec{x}_{k,t}$ in P_t
10. Generate $\vec{g}_{k,t}$ via the global surrogate-assisted search as explained in Section IV-C;
11. Evaluate $\vec{g}_{k,t}$ with the original objective function and constraints: $f(\vec{g}_{k,t}), g_1(\vec{g}_{k,t}), \ldots, g_p(\vec{g}_{k,t})$;
12. Update archive: $A = A \cup (\vec{g}_{k,t}, f(\vec{g}_{k,t}), g_1(\vec{g}_{k,t}), \ldots, g_p(\vec{g}_{k,t}))$;
13. $FEs = FEs + 1$;
14. Compare $\vec{g}_{k,t}$ with $\vec{x}_{k,t}$ according to the feasibility rule, and store the better one into Q_t;
15. **End For**
    // The local surrogate-assisted phase
16. **For** each individual $\vec{x}_{k,t}$ in Q_t
17. Generate $\vec{l}_{k,t}$ via the local surrogate-assisted search as explained in Section IV-D;
18. Evaluate $\vec{l}_{k,t}$ with the original objective function and constraints: $f(\vec{l}_{k,t}), g_1(\vec{l}_{k,t}), \ldots, g_p(\vec{l}_{k,t})$;
19. Update archive: $A = A \cup (\vec{l}_{k,t}, f(\vec{l}_{k,t}), g_1(\vec{l}_{k,t}), \ldots, g_p(\vec{l}_{k,t}))$;
20. $FEs = FEs + 1$;
21. Compare $\vec{l}_{k,t}$ with $\vec{x}_{k,t}$ according to the feasibility rule, and store the better one into P_{t+1};
22. **End For**
23. $t = t + 1$;
24. **End While**

Fig. 1. Framework of GLoSADE.

[1] There are two ways to approximate constraints: 1) approximating all the constraints by making use of the degree of constraint violation $G(\vec{x})$; and 2) approximating each of the constraints. The former can easily compute the feasibility of an individual; however, it cannot identify the structure of each constraint and reflect the property of the boundaries of the feasible region. As a result, we choose the latter in this paper.

1. **If** *rand*<0.5 // *rand* is a uniformly distributed random number between 0 and 1
2. Generate λ mutant vectors using DE/rand/1 in Eq. (3): $\vec{v}_{1,t},\ldots,\vec{v}_{\lambda,t}$;
3. Implement the binomial crossover of DE in Eq. (7) on $\vec{x}_{k,t}$ and $\vec{v}_{i,t}(i=1,\ldots,\lambda)$ to generate λ trial vectors: $\vec{u}_{1,t},\ldots,\vec{u}_{\lambda,t}$;
4. Evaluate these λ trial vectors with GRNN, and select the best one based on the feasibility rule, denoted as $\vec{g}_{k,t}$;
5. **Else**
6. Generate λ trial vectors using DE/current-to-rand/1 in Eq. (5): $\vec{u}_{1,t},\ldots,\vec{u}_{\lambda,t}$;
7. Calculate the uncertainty values of these λ trial vectors according to Eq. (14), and choose the one with the highest uncertainty, denoted as $\vec{g}_{k,t}$;
8. **End If**

Fig. 2. Global surrogate-assisted search.

expensive RBF to construct multiple local surrogates. The training time complexity of GRNN and RBF is $O(DN)$ and $O(DN^3)$ respectively, where N denotes the size of training data. GRNN is a regression model which does not model the training data exactly but gets as close as possible to the training data on average; thereby it is capable of smoothing the rugged fitness landscape. In contrast, RBF is an interpolation model. During the modeling, RBF passes through each of the training data; hence it is able to approximate the fitness landscape accurately. As a result, by combining GRNN in the global surrogate-assisted phase and RBF in the local surrogate-assisted phase, this paper strikes a balance between the "curse of uncertainty" and the "blessing of the uncertainty".

As shown in Step 8) of Fig. 1, all the individuals in archive A are used to construct a GRNN in the global surrogate-assisted phase. It is because we would like to identify the fitness landscape in the whole search space by taking advantage of all the historical data. Since local search is performed on each individual, a more reasonable way is to construct a RBF for each individual in the local surrogate-assisted phase.

C. Global Surrogate-Assisted Search

The global surrogate-assisted search is a surrogate-assisted prescreening method. For each target vector $\vec{x}_{k,t}$ in P_t, it involves two strategies to produce multiple trial vectors, evaluate these trial vectors, and select the best candidate $\vec{g}_{k,t}$ from these trial vectors. These two strategies are applied with the same probability (i.e., 0.5).

In the first strategy, DE/rand/1/bin is adopted to generate λ trial vectors for each target vector in the population, because of its good exploration capability. Afterward, GRNN evaluates these trial vectors and the best candidate is selected according to the feasibility rule [17]. The feasibility rule compares pairwise individuals as follows:

- When comparing two solutions that are predicted to be feasible, the one with the better predicted objective function value is chosen.
- When comparing a solution that is predicted to be feasible with a solution that is predicted to be infeasible, the former is chosen.
- When comparing two solutions that are predicted to be infeasible, the one with the lower predicted degree of constraint violation is chosen.

It is evident that if the trial vectors only contain infeasible solutions, the feasibility rule selects the best infeasible solution to guide the population toward the feasible region, which is the first promising region needed to be located. On the other hand, if the trial vectors contain at least one feasible solution, the feasibility rule selects the best feasible solution so as to guide the population toward the region containing the optimum, which is another promising region needed to be located.

In the first strategy, firstly, DE/rand/1/bin explores the search space by generating multiple trial vectors. Subsequently, GRNN evaluates these trial vectors and filters some trial vectors in the unpromising region because GRNN can smooth the rugged fitness landscape. Lastly, the feasibility rule selects the best candidate among the trial vectors. Thus, through the above steps, the first strategy achieves the quick locating of the promising region.

Since the computation of the original objective function and constraints is extremely expensive in ECOPs, it is desirable to evaluate the individuals with the original objective function and constraints as less as possible, which results in the limited training data for modeling. Obviously, the above phenomenon has a side effect on the quality of a surrogate. It is noteworthy that the surrogate with the wrong global optimum will definitely mislead the evolutionary search. In order to prevent the above phenomenon, an uncertainty-based rule is proposed in this paper to improve the quality of surrogates including GRNN and RBF. It selects a candidate with the highest uncertainty from the trial vectors. The uncertainty-based rule is inspired by the concept of active learning [54], where the training data are selected actively to improve the quality of a surrogate. The uncertainty measures the prediction confidence of a surrogate at a given point. The commonly used way to measure the uncertainty is the variance of the prediction value. The larger the variance, the more sparse the sample points around a given point, so the more inaccurate the prediction of a surrogate at this point. Among the regression models, the Bayesian regression [55] can calculate the variance of a surrogate. Note, however, that the Bayesian regression contains some problem-dependent hyper-parameters. Therefore, to overcome the dependence of problem-specific prior knowledge, we assume that the noise error of the Bayesian regression is zero and choose the cubic form function as the basis function. Finally, the variance is computed in the form of RBF as follows:

$$\sigma^2(\vec{x}) = -\vec{\phi}(\vec{x})\Phi^{-1}\vec{\phi}(\vec{x})^T \tag{14}$$

where $\vec{\phi}(\vec{x}) = (\phi(\vec{x}-\vec{x}_1),\ldots,\phi(\vec{x}-\vec{x}_N))$ and Φ has the same definition as in Eq. (13). In this paper, Eq. (14) is used to measure the uncertainty of a surrogate at a given point. To save the computational time, the uncertainty of a surrogate in $\vec{x}_{k,t}$ is computed with the η nearest points in archive A to $\vec{x}_{k,t}$. In this paper η=100.

In the second strategy, DE/current-to-rand/1 is used to

generate λ trial vectors. It is necessary to emphasize that the binomial crossover is not applied to DE/current-to-rand/1 and it thus exhibits the rotation-invariant feature [56]. In DE/current-to-rand/1, each target vector $\vec{x}_{k,t}$ in P_t learns the information from other individuals randomly. Moreover, the center of all the trial vectors is the target vector, which means that all the trial vectors are distributed around the target vector. Thus, the trial vector with the highest uncertainty has the most potential to enhance the quality of surrogates in the neighborhood of the target vector. At the early stage of evolution, the trial vectors in the population scatter in the search space and the diversity of the population is good. Under this condition, the trial vectors with the highest uncertainty can be used to improve the quality of surrogates in the entire search space. In addition, at the middle and later stages of evolution, the target vectors in the population gather in a small search region near the optimal solution. Accordingly, the trial vectors chosen by the uncertainty-based rule has the capability to refine the quality of surrogates in the neighborhood of the optimal solution. Therefore, combining DE/current-to-rand/1 with the uncertainty-based rule is able to improve the quality of surrogates over the course of evolution.

The implementation of the global surrogate-assisted search is shown in Fig. 2.

D. Local Surrogate-Assisted Search

To further accelerate the convergence, the interior point method [57], a popular local search method, is incorporated into the local surrogate-assisted search. For each individual $\vec{x}_{k,t}$ in Q_t, the interior point method solves an optimization problem with the following form:

$$\begin{aligned} &\text{minimize: } \hat{f}(\vec{x}) \\ &\text{subject to: } \hat{g}_j(\vec{x}) \le 0,\ j = 1,...,p \\ &\qquad \tilde{L}_i \le x_i \le \tilde{U}_i, i = 1,\ldots,D \end{aligned} \tag{15}$$

where $\hat{f}(\vec{x})$ and $\hat{g}_j(\vec{x})$ are the RBF surrogates built with the K nearest points in archive A to $\vec{x}_{k,t}$ for the original objective function and the jth constraint, respectively, and $\tilde{L}_i$ and $\tilde{U}_i$ are the minimum and maximum values of the ith dimension of the K nearest points in archive A to $\vec{x}_{k,t}$, respectively. In this paper, $K = \max((D+1)(D+2)/2, 100)$. [2]

In the local surrogate-assisted search, the interior point method iteratively minimizes the optimization problem in Eq. (15) on the approximated landscape, starting with the initial solution $\vec{y}_0 (\vec{y}_0 = \vec{x}_{k,t})$. In the rth iteration, a gradient descent direction $\vec{p}_r$ and a step size α_r are constructed by making use of $\hat{f}(\vec{x}), \hat{g}_1(\vec{x}), \ldots, \hat{g}_p(\vec{x})$. As a result, a new solution $\vec{y}_{r+1}$ is generated: $\vec{y}_{r+1} = \vec{y}_r + \alpha_r \vec{p}_r$. The interior point method ends when 300 iterations are exhausted. With the termination of the interior point method, the resultant solution (denoted as $\vec{l}_{k,t}$) is considered as the best candidate generated by the local surrogate-assisted search.

Remark 1: Recently, researchers have proposed several surrogate-assisted DE for tackling EUOPs [59], [60]. However, to the best of knowledge, this paper presents the first attempt to integer DE with surrogates for dealing with ECOPs. Moreover, in this paper we suggest a global and local surrogate-assisted DE framework.

V. Experimental Study

The benchmark test functions with only inequality constraints collected in IEEE CEC2006 [27], IEEE CEC2010 [28], and IEEE CEC2017 [29] were employed to demonstrate the capability of GLoSADE. We assume that the computation of their objective functions and constraints is expensive in this empirical study. The main characteristics of these test functions are reported in Table S1, Table S2, and Table S3 in the supplementary file.

GLoSADE includes five main parameters: the population size (*NP*), the number of trial vectors generated for each target vector in the global surrogate-assisted search (λ), the scaling factor and the crossover control parameter of DE/rand/1/bin (F_1 and CR_1), and the scaling factor of DE/current-to-rand/1 (F_2). These parameters were set as follows: NP=80, λ=100, F_1=0.8, CR_1=0.4, and F_2=0.4. For each test function, 25 independent runs were executed.

A. Comparison with mViE on Test Functions Collected in IEEE CEC2006

Firstly, the performance of GLoSADE was compared with that of mViE [61] on 13 benchmark test functions from IEEE CEC2006. mViE is an outstanding method recently proposed by Andrea *et al.* for solving constrained optimization problems. In [61], Andrea *et al.* compared mViE with several popular and representative methods in constrained evolutionary optimization, such as ε-DE [62], ε-RDE [63], PSO [64], ASRES [65], (μ+λ)-CDE [66], and ICDE [67]. The experimental results confirm that mViE outperforms them. For the above compared methods, ε-RDE and ASRES belong to SAEAs. Note that the main advantage of mViE is its efficiency. As reported in [61], mViE can find the optimal solutions within a very limited number of FEs for these 13 benchmark test functions.

Two performance metrics were used to compare GLoSADE with mViE: [3]

- The first performance metric is the successful rate (SR), which represents the percentage of the successful runs. A run is considered as successful if it can find a feasible solution satisfying the successful condition ($f(\vec{x}_{best}) - f(\vec{x}^*) < 0.0001$) within the given maximum number of FEs (*MaxFEs*), where $\vec{x}_{best}$ is the best feasible solution found and $\vec{x}^*$ is the best known solution. In the experiments, *MaxFEs* in GLoSADE was set to 7000 for

[2] As pointed out in [58], when establishing a quadratic interpolation model, at least $(D+1)(D+2)/2$ sample points are required to determine its parameters. As we know, the quadratic interpolation model is a basic nonlinear model. Therefore, when we build a more complex model, it is natural that the number of sample points should be greater than or equal to $(D+1)(D+2)/2$.

[3] To ensure a fair comparison, the experimental results of mViE were directly taken from the original paper [59].

TABLE I
EXPERIMENTAL RESULTS OF MVIE AND GLOSADE IN TERMS OF SR, SFES, AND AR OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS WITH ONLY INEQUALITY CONSTRAINTS FROM IEEE CEC2006

Problem	SR		Mean SFEs±Std Dev		AR
	mViE	GLoSADE	mViE	GLoSADE	
$CEC2006_{01}$	100%	100%	2.06E+04±2.75E+03−	3.68E+02±8.37E+02	98.22%
$CEC2006_{02}$	100%	100%	6.80E+04±3.08E+04−	3.15E+04±2.85E+03	53.60%
$CEC2006_{04}$	100%	100%	4.54E+03±2.15E+03−	5.12E+02±7.52E+02	88.72%
$CEC2006_{06}$	100%	100%	2.78E+03±1.88E+03−	6.48E+02±1.41E+02	76.71%
$CEC2006_{07}$	100%	100%	1.05E+04±8.73E+03−	9.04E+02±1.40E+02	91.40%
$CEC2006_{08}$	100%	100%	5.04E+02±1.83E+02≈	4.80E+02±9.71E+01	4.76%
$CEC2006_{09}$	100%	100%	5.19E+03±5.06E+03−	2.88E+03±1.60E+02	44.54%
$CEC2006_{10}$	100%	100%	2.39E+04±2.29E+04−	2.17E+03±5.34E+02	90.89%
$CEC2006_{12}$	100%	100%	3.97E+03±2.37E+03−	1.45E+03±1.65E+03	63.30%
$CEC2006_{16}$	100%	100%	3.84E+03±2.10E+03−	1.30E+03±5.14E+02	66.07%
$CEC2006_{18}$	100%	100%	1.39E+04±1.78E+04−	1.50E+03±5.46E+02	89.19%
$CEC2006_{19}$	100%	100%	2.68E+04±3.03E+03−	2.46E+03±3.31E+02	90.80%
$CEC2006_{24}$	100%	100%	8.38E+02±4.00E+02−	4.00E+02±0.00E+00	52.27%
				Mean AR	70.04%

all the test functions expect for $CEC2006_{02}$. Due to the fact that it is very difficult to find the optimal solution of $CEC2006_{02}$, the corresponding *MaxFEs* was set to 40000.

- For the second performance metric, the number of FEs required to reach the successful condition is recorded (denoted as SFEs) in each run, which is used to assess the convergence speed. Furthermore, the acceleration rate (AR) between mViE and GLoSADE is calculated for each test function:

$$\mathrm{AR} = \frac{\mathrm{MSFEs}_{\mathrm{mViE}} - \mathrm{MSFEs}_{\mathrm{GLoSADE}}}{\mathrm{MSFEs}_{\mathrm{mViE}}} \times 100\% \qquad (16)$$

where $\mathrm{MSFEs}_{\mathrm{mViE}}$ and $\mathrm{MSFEs}_{\mathrm{GLoSADE}}$ denote the mean SFEs derived from mViE and GLoSADE, respectively.

Table I summarizes the statistics of SR and SFEs provided by mViE and GLoSADE. As shown in Table I, both mViE and GLoSADE can successfully solve these 13 test functions in all 25 runs. Overall, GLoSADE can succeed in satisfying the successful condition within a small number of FEs. Specifically, for six test functions (i.e., $CEC2006_{01}$, $CEC2006_{04}$, $CEC2006_{06}$, $CEC2006_{07}$, $CEC2006_{08}$, and $CEC2006_{24}$), GLoSADE takes less than 1000 FEs on average to reach the global optimum at the precision of 0.0001. For three test functions (i.e., $CEC2006_{12}$, $CEC2006_{16}$, and $CEC2006_{18}$), GLoSADE consumes less than 1500 FEs on average to satisfy the successful condition. It is worth noting that $CEC2006_{18}$ contains 13 nonlinear constraints and the feasibility ratio is approximate to zero, which reveals that GLoSADE has the capability to solve ECOPs with many nonlinear constraints and extremely small feasible region. For three test functions (i.e., $CEC2006_{09}$, $CEC2006_{10}$, and $CEC2006_{19}$), the mean SFEs is less than 2900. $CEC2006_{02}$ is a high-dimensional constrained optimization problem with one highly nonlinear objective function and one highly nonlinear constraint. Moreover, $CEC2006_{02}$ contains many local optima. Owing to the high dimensionality, high nonlinearity, and limited training data, it is very likely to build a surrogate with the wrong global optimum for $CEC2006_{02}$, thus misleading the evolutionary search. Although $CEC2006_{02}$ is such a complex constrained optimization problem, GLoSADE consistently achieves the successful condition with not very big SFEs in 25 runs.

To detect the statistical difference between mViE and GLoSADE, the *t*-test at a 0.05 significance level was performed based on the average SFEs. In Table I, "+", "−", and "≈" denote that the performance of mViE is better than, worse than, and similar to that of GLoSADE, respectively. It can be seen from Table I that GLoSADE performs significantly better than mViE on all the test functions except for $CEC2006_{08}$. As far as AR is concerned, GLoSADE reduces on average 70.04% FEs to reach the optimal solution, compared with mViE. Moreover, on six test functions (i.e., $CEC2006_{01}$, $CEC2006_{04}$, $CEC2006_{07}$, $CEC2006_{10}$, $CEC2006_{18}$, and $CEC2006_{19}$), GLoSADE converges on average more than 80% faster than mViE toward the optimal solution. In particular, for $CEC2006_{01}$, GLoSADE saves the number of FEs up to 98.22% compared with mViE.

The above comparison suggests that GLoSADE shows better convergence speed than mViE.

B. Comparison with (μ+μ)-CEP-RBF and FROFI on Test Functions Collected in IEEE CEC2006, IEEE CEC2010, and IEEE CEC2017

Subsequently, we compared GLoSADE against (μ+μ)-CEP-RBF [43] and FROFI [11] on 13 test functions from IEEE CEC2006 [27], six test functions with 10 dimensions (10D) and 30 dimensions (30D) from IEEE CEC2010 [28], and nine test functions with 10D and 30D from IEEE CEC2017 [29]. (μ+μ)-CEP-RBF is a RBF-assisted evolutionary programming proposed by Regis for high-dimensional black-box ECOPs. (μ+μ)-CEP-RBF has demonstrated its excellent performance in solving some benchmark and practical high-dimensional black-box ECOPs. In addition, FROFI is a very recent constrained optimization evolutionary algorithm, which incorporates objective function information into the feasibility rule.

To compare the performance of GLoSADE with that of (μ+μ)-CEP-RBF and FROFI, we re-implemented the codes of (μ+μ)-CEP-RBF and FROFI provided by the authors in [43] and [11], respectively. All the parameter settings of (μ+μ)-CEP-RBF and FROFI were consistent with the original papers. It is necessary to emphasize that in [43] the initial population must have at least one feasible individual. In contrast, GLoSADE

TABLE II
EXPERIMENTAL RESULTS OF (μ+μ)-CEP-RBF, FROFI, AND GLOSADE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS WITH ONLY INEQUALITY CONSTRAINTS FROM IEEE CEC2006. (#) DENOTES THE NUMBER OF TRIALS IN WHICH AT LEAST ONE FEASIBLE SOLUTION IS FOUND IN THE FINAL POPULATION OVER 25 INDEPENDENT RUNS.

Problem	Mean Function Error Value±Std Dev			SR		
	(μ+μ)-CEP-RBF	FROFI	GLoSADE	(μ+μ)-CEP-RBF	FROFI	GLoSADE
$CEC2006_{01}$	6.48E-01±9.16E-01−	9.86E+00±1.18E+00−	4.73E-07±5.29E-07	40%	0%	100%
$CEC2006_{02}$	4.88E-01±5.16E-02−	4.69E-01±2.89E-02−	4.09E-01±2.74E-02	0%	0%	0%
$CEC2006_{04}$	1.13E-06±2.51E-06−	1.30E+02±3.38E+01−	2.27E-07±1.10E-07	100%	0%	100%
$CEC2006_{06}$	5.86E-03±1.28E-02−	2.35E+02±1.08E+02−	8.13E-06±1.91E-05	60%	0%	100%
$CEC2006_{07}$	4.00E-01±4.74E-01−	9.52E+01±3.30E+01−	8.13E-08±3.43E-08	0%	0%	100%
$CEC2006_{08}$	1.33E-02±2.98E-02−	6.38E-05±7.03E-05−	8.85E-11±1.45E-11	80%	100%	100%
$CEC2006_{09}$	1.04E+02±1.91E+02−	7.04E+01±2.71E+01−	3.82E-07±6.07E-07	0%	0%	100%
$CEC2006_{10}$	1.44E+02±8.59E+01−	(15)	1.71E-06±2.91E-06	0%	0%	100%
$CEC2006_{12}$	1.13E-03±2.52E-03−	2.71E-04±2.35E-04≈	3.04E-04±4.54E-04	60%	24%	84%
$CEC2006_{16}$	1.18E-07±1.80E-07−	1.42E-01±5.16E-02−	8.11E-08±8.85E-09	100%	0%	100%
$CEC2006_{18}$	2.22E-01±7.61E-02−	(0)	9.66E-05±3.04E-04	0%	0%	100%
$CEC2006_{19}$	6.72E+00±3.44E+00−	3.28E+02±9.17E+01−	1.03E-05±1.46E-05	0%	0%	100%
$CEC2006_{24}$	5.98E-11±8.33E-11+	6.16E-03±4.58E-03−	1.50E-08±7.74E-09	100%	0%	100%

does not have any specific requirements for the initial population. For fair comparison, the initial population of (μ+μ)-CEP-RBF, FROFI, and GLoSADE were generated with the same method, i.e., Latin hypercube design. The experimental results of (μ+μ)-CEP-RBF, FROFI, and GLoSADE are summarized in Table II, and Table S4 and Table S5 in the supplementary file under the condition that *MaxFES*=3000. In Table II, Table S4, and Table S5, if a method cannot consistently find at least one feasible solution in the final population over 25 independent runs, we only reported the number of feasible trials. In addition, the Wilcoxon's rank sum test at a 0.05 significance level was used to test the statistical significance between GLoSADE and each of (μ+μ)-CEP-RBF and FROFI.

Next, we give the detailed performance comparison from two aspects:

- For the 13 test functions from IEEE CEC2006, their optimal solutions have been reported in [27]. Thus, Table II records the mean and standard deviation of the function error value ($f(\vec{x}_{best}) - f(\vec{x}^*)$) provided by the three compared methods over 25 runs on these 13 test functions. From Table II, GLoSADE outperforms (μ+μ)-CEP-RBF on all the 13 test functions expect for $CEC2006_{24}$ according to the Wilcoxon's rank sum test at a 0.05 significance level, in terms of the average function error value. For $CEC2006_{24}$, (μ+μ)-CEP-RBF performs better than GLoSADE. Note that the mean function error values of GLoSADE are several orders of magnitude lower than that of (μ+μ)-CEP-RBF on eight test functions (i.e., $CEC2006_{01}$, $CEC2006_{06}$, $CEC2006_{07}$, $CEC2006_{08}$, $CEC2006_{09}$, $CEC2006_{10}$, $CEC2006_{18}$, and $CEC2006_{19}$). This may be because the local surrogate-assisted search of GLoSADE enhances the accuracy of individuals by the interior point method. Moreover, we further computed SR based on the function error value. We can also observe from Table II that GLoSADE provides higher SR than (μ+μ)-CEP-RBF on nine test functions (i.e., $CEC2006_{01}$, $CEC2006_{06}$, $CEC2006_{07}$, $CEC2006_{08}$, $CEC2006_{09}$, $CEC2006_{10}$, $CEC2006_{12}$, $CEC2006_{18}$, and $CEC2006_{19}$). Moreover, for eight out of these nine test functions, GLoSADE is able to provide 100% SR. For the remaining four test functions, GLoSADE provides the same SR with (μ+μ)-CEP-RBF. In addition, FROFI shows the worst performance among the three compared methods. To be specific, it performs worse than GLoSADE on all the 13 test functions with the exception of $CEC2006_{12}$. For $CEC2006_{12}$, there is no significant difference between FROFI and GLoSADE. Moreover, FROFI fails to consistently find feasible solutions for two test functions, i.e., $CEC2006_{10}$ and $CEC2006_{18}$, and successfully solves only one test function, i.e., $CEC2006_{08}$. The poor performance of FROFI signifies that a method which aims at solving the general constrained optimization problems cannot achieve competitive performance on ECOPs due to the limited number of FEs.
- For the six test functions with 10D and 30D from IEEE CEC2010, and the nine test functions with 10D and 30D from IEEE CEC2017, their optimal solutions cannot be known *a priori*. As a result, we recorded the average and standard deviation of objective function values resulting from the three compared methods over 25 runs in Table S4 and Table S5. It is necessary to point out that these test functions include some extremely complex properties: highly nonlinear objective functions and/or constraints, rotated objective function, non-separable decision variables, large search space, etc. Consequently, they pose a grand challenge to the three compared methods. However, as can be seen from Table S4 and Table S5, GLoSADE still shows better overall performance than the two competitors. In terms of the six test functions with 10D and 30D from IEEE CEC2010, GLoSADE performs better than or similar to (μ+μ)-CEP-RBF and FROFI according to the Wilcoxon's rank sum test at a 0.05 significance level. Compared with GLoSADE, (μ+μ)-CEP-RBF and FROFI cannot consistently provide feasible solutions on one case ($CEC2010_{15}$ with 30D) and two cases ($CEC2010_{13}$ with 30D and $CEC2010_{15}$ with 30D), respectively. With respect to the nine test functions with 10D and 30D from IEEE CEC2017, (μ+μ)-CEP-RBF is better than GLoSADE on two cases (i.e., $CEC2017_{01}$

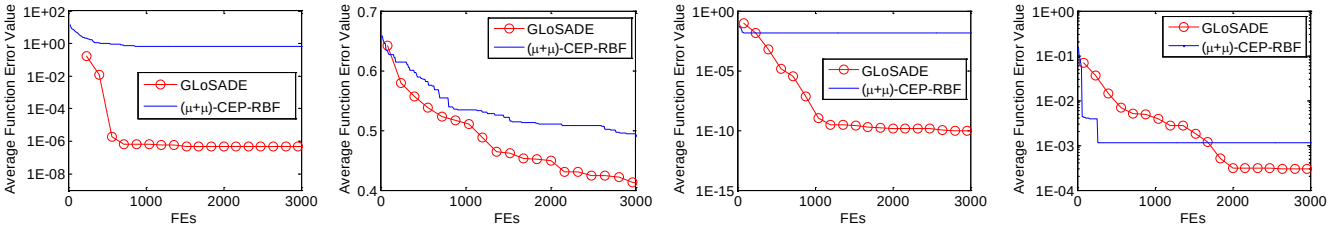

Fig. 3. Evolution of the mean function error values of (μ+μ)-CEP-RBF and GLoSADE on four test functions. (a) $CEC2006_{01}$. (b) $CEC2006_{02}$. (c) $CEC2006_{08}$. (d) $CEC2006_{12}$.

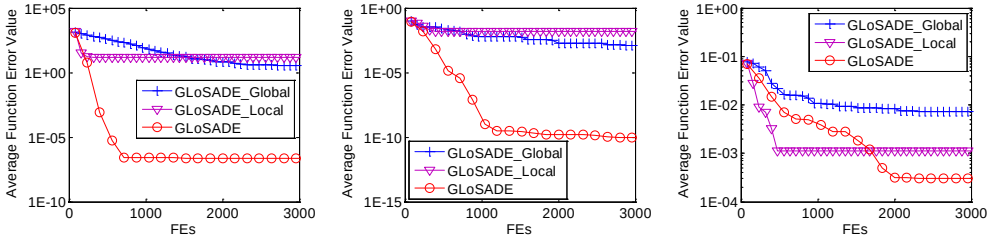

Fig. 4. Evolution of the mean function error values of GLoSADE_Global, GLoSADE_Local, and GLoSADE on three test functions. (a) $CEC2006_{04}$. (b) $CEC2006_{08}$. (c) $CEC2006_{12}$.

with 10D and $CEC2017_{02}$ with 10D) and FROFI has an edge over GLoSADE on one case (i.e., $CEC2017_{01}$ with 10D). However, GLoSADE beats (μ+μ)-CEP-RBF and FROFI on nine and ten cases, respectively. Moreover, FROFI is unable to find any feasible solution on $CEC2017_{13}$ with 10D and $CEC2017_{22}$ with 10D, respectively, compared with GLoSADE.

Fig. 3 plots the convergence curves for GLoSADE and (μ+μ)-CEP-RBF on four representative test functions (i.e., $CEC2006_{01}$, $CEC2006_{02}$, $CEC2006_{08}$, and $CEC2006_{12}$). The horizontal axis represents the number of FEs and the vertical axis represents the average function error value. As shown in Fig. 3, (μ+μ)-CEP-RBF tends to converge very fast in the early stage of evolution. Note, however, that it suffers from stagnation in the middle and later stages of evolution. In contrast, GLoSADE keeps evolving forward in the middle and later stages of evolution. The above phenomenon can be explained as follows. In (μ+μ)-CEP-RBF, the best candidate is preselected from a huge number of trial offspring (i.e., $\min(1000D,10000)$ trial offspring), which leads to the dramatic decrease of diversity of the population. Whereas, GLoSADE selects the best candidate from just 100 trial offspring. Moreover, the uncertainty-based rule of GLoSADE is also beneficial to increase the diversity of the population.

The above comparison demonstrates that GLoSADE outperforms (μ+μ)-CEP-RBF and FROFI in terms of the solution quality on three test suites.

C. Effectiveness of Some Components in GLoSADE

In this subsection, additional experiments were conducted to investigate the effectiveness of some components in GLoSADE. The 13 test functions from IEEE CEC2006 were chosen to produce the experimental results over 25 independent runs with *MaxFES*=3000. Moreover, the Wilcoxon's rank sum test at a 0.05 significance level was performed between GLoSADE and each competitor.

1) Effectiveness of the Uncertainty-Based Rule: The uncertainty-based rule is designed in this paper to alleviate the inaccuracy of surrogates. To study the effectiveness of this rule, we considered another variant of GLoSADE, called GLoSADE_WoU, in which the uncertainty-based rule was removed from GLoSADE. The mean and standard deviation of function error values resulting from GLoSADE and GLoSADE_WoU are summarized in Table S6 of the supplementary file. As shown in Table S6, GLoSADE surpasses GLoSADE_WoU on nine test functions. In contrast, GLoSADE_WoU is better than GLoSADE on only one test function (i.e., $CEC2006_{09}$). The reasons why the performance of GLoSADE_WoU is better than GLoSADE on $CEC2006_{09}$ are twofold: 1) the convergence speed of both GLoSADE_WoU and GLoSADE is very fast for $CEC2006_{09}$; and 2) the uncertainty-based rule in GLoSADE aims at putting more emphasis on the sparse regions and adding the diversity of the population; thus, the accuracy of the solution provided by GLoSADE is slightly worse than that provided by GLoSADE_WoU. In addition, the performance improvement provided by GLoSADE against GLoSADE_WoU is quite significant on seven test functions (i.e., $CEC2006_{06}$, $CEC2006_{07}$, $CEC2006_{08}$, $CEC2006_{10}$, $CEC2006_{12}$, $CEC2006_{18}$, and $CEC2006_{19}$).

The above analysis verifies that the uncertainty-based rule can enhance the quality of the surrogates and thus improve the performance of GLoSADE.

2) Effectiveness of the Global Surrogate-Assisted Phase and the Local Surrogate-Assisted Phase: GLoSADE consists of two important phases: global surrogate-assisted phase and local surrogate-assisted phase. In order to analyze the role of each phase in GLoSADE, two variants of GLoSADE, namely GLoSADE_Global and GLoSADE_Local, were considered. In GLoSADE_Global, the local surrogate-assisted phase was removed from GLoSADE. In addition, in GLoSADE_Local, the global surrogate-assisted phase was eliminated from GLoSADE. Table S7 in the supplementary file summarizes the mean and standard deviation of function error values derived

from GLoSADE_Global, GLoSADE_Local, and GLoSADE. Besides, Fig. 4 shows the convergence curves of the three compared methods on three representative test functions (i.e., $CEC2006_{04}$, $CEC2006_{08}$, and $CEC2006_{12}$).

From Table S7, GLoSADE exhibits superior performance against GLoSADE_Global and GLoSADE_Local on 13 and 11 test functions, respectively. However, GLoSADE_Global and GLoSADE_Local cannot beat GLoSADE on any test functions. More importantly, both GLoSADE_Global and GLoSADE_Local cannot consistently find feasible solutions on one test function. The poor performance of GLoSADE_Global and GLoSADE_Local could be attributed to the following facts: 1) without the local surrogate-assisted phase, GLoSADE_Global suffers from slow convergence speed; and 2) due to the lack of sufficient sampling in the search space, GLoSADE_Local is prone to converge to a local attraction basin very fast and then fall into stagnation during the subsequent evolution. Fig. 4 further verifies the above analysis.

Based on the results provided in the above experiments, we can conclude that both kinds of surrogate-assisted phases are certainly important for the performance of GLoSADE.

3) Effectiveness of GRNN in the Global Surrogate-Assisted Phase: In GLoSADE, following the guideline of "blessing of the uncertainty", GRNN was used to construct the global surrogate in the global surrogate-assisted phase. However, there are also other popular techniques, such as RBF and GP, to fit global surrogates. One may be interested in what will happen to the performance of GLoSADE if we use RBF or GP instead of GRNN in the global surrogate-assisted phase. To this end, we replaced GRNN with RBF and GP, and the resultant methods are denoted as GLoSADE_RBF and GLoSADE_GP. Table S8 in the supplementary file records the experimental results of GLoSADE_RBF, GLoSADE_GP, and GLoSADE.

The first observation from Table S8 is that GLoSADE, GLoSADE_RBF, and GLoSADE_GP show overall similar performance, in terms of the function error value. Specifically, GLoSADE outperforms GLoSADE_RBF and GLoSADE_GP on three and four test functions, respectively, and GLoSADE_RBF and GLoSADE_GP perform better than GLoSADE on two and five test functions, respectively. However, as far as the runtime is concerned, the performance difference among them is quite significant. [4] It can be seen that GLoSADE is on average 1.9 times and 18.5 times faster than GLoSADE_RBF and GLoSADE_GP, respectively. It is because the computational time complexity of RBF and GP is much higher than that of GRNN. To be specific, the computational time complexity of RBF is $O(DN^3)$ as introduced in Section IV-B, and the computational time complexity of a typical learning algorithm for GP is $O(TDN^3)$ [4], where T is the number of iterations, D is the number of decision variables, and N denotes the size of training data.

Therefore, GRNN is a good choice for the global surrogate-assisted phase. It can achieve competitive performance with a less computational cost.

Remark 2: The effect of the parameter settings of GLoSADE was given in Section S-I of the supplementary file.

[4] The runtime was recorded on Intel(R) machine with Core(TM) i5-4590 CPU @3.30GHz and 8GB of RAM.

VI. Real-World Applications

GLoSADE was further applied to solve two practical ECOPs, i.e., the optimal shape design of transonic airfoil and the topology optimization of energy-absorbing component.

A. Optimal Shape Design of Transonic Airfoil

The optimal shape design of transonic airfoil is a complex and expensive task, which needs numerous computational fluid dynamic (CFD) simulations. Herein, we performed an optimal shape design of a 2D transonic airfoil, the initial geometry of which is the NACA0012 airfoil in transonic inviscid flow (shown in Fig. S4 of the supplementary file). The shape of the 2D transonic airfoil is parameterized by a set of Hicks-Henne bump functions based on the initial geometry NACA0012. Hicks-Henne bump functions involve 38 design variables and the range of each design variable is [-0.02, 0.02]. By changing the parameters of Hicks-Henne bump functions, different shapes of the airfoil can be obtained. In this application, the objective is to minimize the drag of the airfoil with lift, moment, and thickness constraints:

$$\begin{aligned} \text{minimize: } & drag(\vec{x}) \\ \text{subject to: } & lift(\vec{x}) > 0.327 \\ & moment(\vec{x}) > 0 \\ & thickness(\vec{x}) > 0.12 \end{aligned} \tag{17}$$

The CFD simulations of aerodynamic performance were undertaken by SU2 [68], which is a piece of open-source software. The resources (i.e., the configuration file and the mesh file) of this application for simulation are available in the directory of SU2. SU2 takes around 14 seconds to run one simulation of the transonic airfoil in our computer.

Fig. S5 in the supplementary file shows the convergence of the mean of the best feasible objective function value for the shape design of 2D transonic airfoil obtained by (μ+μ)-CEP-RBF and GLoSADE after 500 FEs over 25 independent runs. The population size of (μ+μ)-CEP-RBF and GLoSADE was set to 10 in this application. As shown in Fig. S5, GLoSADE exhibits better convergence performance and provides the solution of higher quality. Specifically, the mean drag values resulting from (μ+μ)-CEP-RBF and GLoSADE are 1.61E-06 and 4.05E-07, respectively.

B. Topology Optimization of Energy-absorbing Component

The design of energy-absorbing component in vehicles is an important part of lightweight, and the topological structure of the energy-absorbing component has a significant impact on the crashworthiness of vehicles. Indeed, the design of energy-absorbing component is a very expensive optimization problem. Recently, Sun *et al.* [69] proposed an integer-coded genetic algorithm (ICGA) to design an energy-absorbing component with multi-cell structure. The schematic of the multi-cell structure used in [69] is shown in Fig. 5, where the length of tube (H) is 150 mm, the cross-sectional dimension (L×L) is 80 mm × 80 mm, and the initial thickness of web walls (T) is 1 mm. In addition, Fig. 6 depicts the variable distribution of this multi-cell structure. As shown in Fig. 6, the structure is 1/8

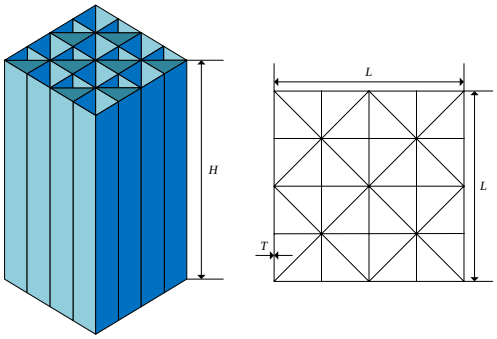

Fig. 5. The geometrical configuration of multi-cell structure.

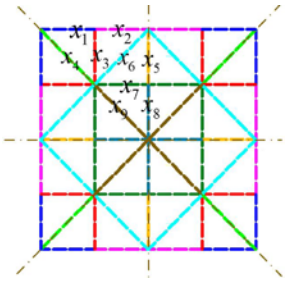

Fig. 6. The variable distribution of multi-cell structure.

Table III
EXPERIMENTAL RESULTS OF ICGA AND GLOSADE OVER FIVE INDEPENDENT RUNS ON TOPOLOGY OPTIMIZATION OF ENERGY-ABSORBING COMPONENT

ICGA Mean Number of FEs±Std Dev	GLoSADE Mean Number of FEs±Std Dev
1.61E+03±2.11E+01	4.12E+02±1.85E+01

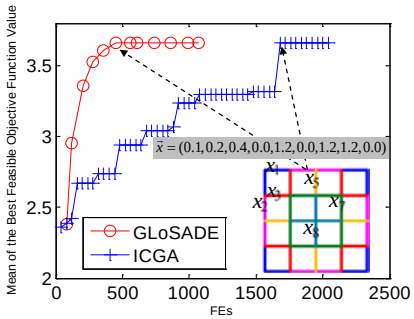

Fig. 7. Convergence curves and the best variable distribution derived from ICGA and GLoSADE.

symmetric; hence, the thicknesses of nine web walls are considered as the design variables: $x_1,\ldots,x_9$, where $x_i \in \{0.0, 0.1,\ldots,1.2\}$ $(i=1,\ldots,9)$. In the design of energy-absorbing component, the energy absorption (*EA*) and the peak crushing force (*PCF*) are commonly used to assess the structural crashworthiness, where *EA* and *PCF* represent the total absorbed energy and the maximum crushing force during the entire collapse process, respectively. The more the energy absorbed, the better the structural crashworthiness. Overall, this design can be formulated to maximize *EA* with the *PCF* and mass constraints:

$$\begin{aligned} &\text{maximize: } EA(\vec{x}) \\ &\text{subject to: } PCF(\vec{x}) \le 55 \\ &\qquad\qquad M(\vec{x}) \le 31.3137 \end{aligned} \tag{18}$$

The nonlinear explicit finite element code, LS-DYNA, was utilized to simulate the crashing process of the multi-cell structure. It needs about 23 minutes to run one simulation of the crashing process of the multi-cell structure in our computer. Due to the fact that this application is extremely computationally expensive, ICGA and GLoSADE were independently run five times. According to the report in [69], the maximum *EA* found by ICGA is 3.66. Table III presents the average number of FEs needed by ICGA and GLoSADE to reach this *EA* value over five runs. From Table III, ICGA consumes on average 1610 FEs. In contrast, GLoSADE only takes on average 412 FEs. Thus, GLoSADE saves 74.4% FEs and reduces (1610-412)*23/60≈459.2 hours compared with ICGA. Fig. 7 plots the convergence curves and the best variable distribution derived from ICGA and GLoSADE.

The above experiments reveal that GLoSADE could be an effective tool to solve ECOPs in the real-world applications.

VII. CONCLUSION

Combination of global search and local search has been proven to be very successful in solving different kinds of optimization problems in the evolutionary computation research community [24], [70]. However, few attempts have been made along this line to investigate the solution of ECOPs, in which the evaluation of the objective function and constraints could be extremely computationally expensive.

This paper attempts to introduce a novel global and local surrogate-assisted DE for solving ECOPs with inequality constraints, called GLoSADE. It consists of two main phases: global surrogate-assisted phase and local surrogate-assisted phase. In the global surrogate-assisted phase, GRNN is used to construct the global surrogates. Afterward, DE/rand/1/bin is combined with the feasibility rule to locate the promising area quickly, and DE/current-to-rand/1 is integrated with the uncertainty-based rule to improve the quality of surrogates to a certain degree. In addition, the local surrogate-assisted phase makes use of RBF to build the local surrogates and the interior point method to refine the quality of each individual, with the aim of speeding up the convergence. Overall, GLoSADE reaches a tradeoff between effectiveness and efficiency.

From the comparative study on benchmark test functions chosen from IEEE CEC2006, IEEE CEC2010, and IEEE CEC2017, the performance GLoSADE is better than that of three other well-known methods in terms of the selected performance metrics. Moreover, GLoSADE has been applied to two real-world cases to verify its capability.

Currently, very few methods provide the experimental results for ECOPs with equality constraints. It is because it is very hard for surrogates to approximate equality constraints, in particular, nonlinear equality constraints. In principle, when dealing with ECOPs, the aim of surrogates (such as GP, RBF, and artificial neural network) is to approximate objective function and constraints. However, nonlinear equality constraints pose a grand challenge to such approximation. In the future, we will study how to improve the performance of GLoSADE on ECOPs with nonlinear equality constraints. Additionally, the feasibility rule [17] is adopted as the

constraint-handling technique in this paper. Further investigation of other constraint-handling techniques (such as multi-objective optimization-based constraint-handling techniques) under the framework of GLoSADE will be an interesting and promising part of our future work.

The Matlab source code of GLoSADE can be downloaded from Y. Wang's homepage: http://www.escience.cn/people/yongwang1/ index.html

REFERENCES

[1] A. I. J. Forrester, N. W. Bressloff, and A. J. Keane, "Optimization using surrogate models and partially converged computational fluid dynamics simulations," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 462, pp. 2177–2204, 2006.

[2] M. Papadrakakis, N. Lagaros, and Y. Tsompanakis, "Structural optimization using evolution strategies and neural networks," *Comput. Methods Appl. Mech. Engrg.*, vol. 156, pp. 309–333, 1998.

[3] F. Grimaccia, M. Mussetta, and R. E. Zich, "Genetical swarm optimization: self-adaptive hybrid evolutionary algorithm for electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 781–785, 2007.

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

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

[6] R. G. Regis and C. Shoemaker, "Local function approximation in evolutionary algorithms for the optimization of costly functions," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 490–505, 2004.

[7] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou, "Metamodel–assisted evolution strategies," in *Proc. Parallel Problem Solving From Nature VII*, Lecture Notes in Computer Science, vol. 2439, 2002, pp. 361–370.

[8] Z. Zhou, Y. S. Ong, M. H. Lim, and B. S. Lee, "Memetic algorithm using multi-surrogates for computationally expensive optimization problems," *Soft Comput.*, vol. 11, no. 10, pp. 957–971, 2007.

[9] K. Liang, X. Yao, and C. Newton, "Combining landscape approximation and local search in global optimization," in *Proc. Congr. Evol. Comput.*, Piscataway, NJ, 1999, pp. 1514–1520.

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

[11] 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 Trans. Cybern.*, vol. 46, no. 12, pp. 2938-2952, 2016.

[12] G. Wu, W. Pedrycz, P. N. Suganthan, and R. Mallipeddi, "A variable reduction strategy for evolutionary algorithms handling equality constraints," *Applied Soft Computing*, vol. 37, pp. 774-786, 2015.

[13] G. Wu, W. Pedrycz, P. N. Suganthan, and H. Li, "Using variable reduction strategy to accelerate evolutionary optimization, " *Applied Soft Computing*, vol. 61, pp. 283-293, 2017.

[14] S. Benhamida and M. Schoenauer, "ASCHEA: new results using adaptive segregational constraint handling," in *Proc. Congr. Evol. Comput.*, Hawaii, USA, 2002, pp. 884–889.

[15] D. Wcoit, A. Esmith, and D. Mtate, "Adaptive penalty methods for genetic optimization of constrained combinatorial problems." *Inf. J. Comput.*, vol. 8, no. 2, pp. 173–182, 1996.

[16] R. Farmani and J. A. Wright, "Self-adaptive fitness formulation for constrained optimization," *IEEE Trans. Evol. Comput*., vol. 7, no. 5, pp. 445–455, Oct. 2003.

[17] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2–4, pp. 311–338, 2000.

[18] E. Mezura-Montes and C. A. Coello Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," *IEEE Trans. Evol. Comput*., vol. 9, no. 1, pp. 1–17, Feb. 2005.

[19] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, 2000.

[20] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Trans. Syst. Man, Cybern. C*, vol. 35, no. 2, pp. 233–243, 2005.

[21] C. A. Coello Coello, "Treating constraints as objectives for single-objective evolutionary optimization," *Eng. Optim.*, vol. 32, no. 2, pp. 275–308, 2000.

[22] C. A. Coello Coello, "Constraint-handling using an evolutionary multiobjective optimization technique," *Civ. Eng. Environ. Syst.*, vol. 17, no. 4, pp. 319–346, 2000.

[23] Z. Cai, and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 658–675, 2006.

[24] Y. Wang, Z. Cai, G. Guo, and Y. Zhou, "Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems," *IEEE Trans. Syst. man, Cybern. B, Cybern.*, vol. 37, no. 3, pp. 560–575, 2007.

[25] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 80–92, Feb. 2008.

[26] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature inspired numerical optimization: past, present and future," *Swarm Evol. Comput*., vol. 1, no. 4, pp. 173–194, 2011.

[27] J. J. Liang, T. P. Runarsson, M. Clerc, P. N. Suganthan, C. A. Coello Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., 2006.

[28] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," Technical Report, Nanyang Technological University, Singapore, 2010.

[29] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization," Technical Report, Nanyang Technological University, Singapore, November 2016.

[30] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[31] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 329, pp. 329-345, 2016.

[32] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Information Sciences*, vol. 423, pp. 172-186, 2018.

[33] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, vol. 2, no. 6, pp. 568–576, 1991.

[34] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Stat.*, vol. 33, no. 3, pp. 1065–1076, 1962.

[35] S. M. Wild and C. Shoemaker, "Global convergence of radial basis function trust region derivative-free algorithms," *SIAM J. Optim.*, vol. 55, no. 3, pp. 1–27, 2013.

[36] E. Krempser, H. S. Bernardino, H. J. C. Barbosa, and A. C. C. Lemonge, "Differential evolution assisted by surrogate models for structural optimization problems," in *Proc. The Eighth International Conference on Engineering Computational Technology*, Stirlingshire, Scotland, 2012, pp. 1–19.

[37] E. Mezura-Montes, L. M. Dávila, and C. A. Coello Coello, "A preliminary study of fitness inheritance in evolutionary constrained optimization," in *Nature Inspired Cooperative Strategies for Optimization*, N. Krasnogor, G. Nicosia, M. Pavone, and D. Pelta, Eds. Berlin, Germany: Springer, 2008, pp. 1–14.

[38] T. P. Runarsson, "Constrained evolutionary optimization by approximate ranking and surrogate models," in *Proc. Parallel Problem Solving From Nature VII*, Lecture Notes in Computer Science, vol. 3242, 2004, pp. 401–410.

[39] D. Büche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with Gaussian process fitness function models," *IEEE Trans. Syst. Man, Cybern. C*, vol. 35, no. 2, pp. 1–12, 2004.

[40] V. Torczon and M. W. Trosset, "Using approximations to accelerate engineering design optimization," in *Proc. 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Amer. Inst. Aero. Astron., Virginia, USA, 1998, pp. 738–748.

[41] M. T. M. Emmerich, K. C. Giannakoglou and B. Naujoks, "Single-objective and multiobjective evolutionary optimization assisted by

Gaussian random field metamodels," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, 2006.
[42] M. Schonlau, W. Welch, and D. Jones, "Global versus local search in constrained optimization of computer models," in *New Developments and Applications in Experimental Design*, N. Flournoy, W. Rosenberger, and W. Wong, Eds. Hayward, CA: Institute of Mathematical Statistics, 1998, vol. 34, pp. 11–25.
[43] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis function," *IEEE Trans. Evol. Comput*., vol. 18, no. 3, pp. 326–347, 2014.
[44] R. G. Regis, "Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points," *Engineering Optimization*, vol. 46, no. 2, pp. 218–243, 2014.
[45] S. Bagheri, W. Konen, C. Foussette, P. Krause, T. Back, and P. Koch, "SACOBRA: Self-adjusting constrained black-box optimization with RBF," In *Proc. 25. Workshop Computational Intelligence*, Dortmund, 2015, pp. 87–96.
[46] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *AIAA J.*, vol. 41, no. 4, pp. 687–696, 2003.
[47] C. T. Lawrence and A. L. Tits, "A computationally efficient feasible sequential quadratic programming algorithm," *SIAM J. Optim.*, vol. 11, no. 4, pp. 1092–1118, 2001.
[48] C. K. Goh, D. Lim, L. Ma, Y. S. Ong, and P. S. Dutta, "A surrogate-assisted memetic co-evolutionary algorithm for expensive constrained optimization problems," in *Proc. CEC*, New Orleans, USA, 2011, pp. 744–749.
[49] S. D. Handoko, C. K. Kwoh, and Y. S. Ong, "Classification-assisted memetic algorithms for equality constrained optimization problems," in *Proc. AI 2009: Advances in Artificial Intelligence*, Melbourne, Australia, 2009, pp. 391–400.
[50] S. D. Handoko, C. K. Kwoh, and Y. S. Ong, "Feasibility structure modeling: an effective chaperone for constrained memetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 740–758, 2010.
[51] R. G. Regis, "Trust regions in surrogate-assisted evolutionary programming for constrained expensive black-box optimization," In: R. Datta and K. Deb (eds), *Evolutionary Constrained Optimization*, 2015, pp. 51–94.
[52] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.
[53] D. Lim, Y. Jin, Y. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, 2010.
[54] D. J. C. MacKay, "Information-based objective functions for active data selection," *Neural Comput.*, vol. 4, no. 4, pp. 590–604, 1992.
[55] C. E. Rasmussen and C. K. I. Williams, "*Gaussian Processes for Machine Learning*," Cambridge, MA: MIT Press, 2006.
[56] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55-66, 2011.
[57] F. A. Potra and S. J. Wright, "Interior-point methods," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 281–302, 2000.
[58] R. G. Regis and C. A. Shoemaker, "A stochastic radial basis function method for the global optimization of expensive functions," *INFORMS Journal on Computing*, vol. 19, no. 4, pp. 497-509, 2007.
[59] S. M. Elsayed, T. Ray, and R.A. Sarker, "A surrogate-assisted differential evolution algorithm with dynamic parameters selection for solving expensive optimization problems," in *Proc. CEC*, Beijing, China, 2014, pp. 1062–1068.
[60] R. Mallipeddi and M. Lee, "An evolving surrogate model-based differential evolution algorithm," *Applied Soft Computing*, vol. 34, pp. 770–787, 2015.
[61] A. Maesani, G. Iacca, and D. Floreano, "Memetic viability evolution for constrained optimization," *IEEE Trans. Evol. Comput*., vol. 20, no. 1, pp. 125–144, 2016.
[62] T. Takahama and S. Sakai, "Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation," in *Proc. CEC*, Barcelona, Spain, 2010, pp. 1680–1688.
[63] T. Takahama and S. Sakai, "Efficient constrained optimization by the ε constrained rank-based differential evolution," in *Proc. CEC*, Brisbane, QLD, Australia, 2012, pp. 1–8.
[64] K. Zielinski and R. Laur, "Constrained single-objective optimization using particle swarm optimization," in *Proc. CEC*, Vancouver, BC, Canada, 2006, pp. 443–450
[65] T. P. Runarsson, "Approximate evolution strategy using stochastic ranking," in *Proc. CEC*, Vancouver, BC, Canada, 2006, pp. 745–752.
[66] Y. Wang and Z. Cai, "Constrained evolutionary optimization by means of (μ+λ)-differential evolution and improved adaptive trade-off model," *Evol. Comput.*, vol. 19, no. 2, pp. 249–285, 2011.
[67] G. Jia, Y. Wang, Z. Cai, and Y. Jin, "An improved (μ+λ)-constrained differential evolution for constrained optimization," *Inform. Sciences*, vol. 222, pp. 302–322, Feb. 2013
[68] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, "SU2: An open-source suite for multiphysics simulation and design," *AIAA J*., vol. 54, no. 3, pp. 828–846, 2016.
[69] G. Sun, T. Liu, J. Fang, G. P. Steven, and Q. Li, "Configurational optimization of multi-cell topologies for multiple oblique loads," *Structural and Multidisciplinary Optimization*, vol. 57, no. 2, pp. 469–488, 2017.
[70] Y. S. Ong, M. H. Lim, and X. S. Chen, "Research frontier: Memetic computation - Past, present & future," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24 -36, 2010.

Yong Wang (M'08-SM'17) received the B.S. degree in automation from the Wuhan Institute of Technology, Wuhan, China, in 2003, and the M.S. degree in pattern recognition and intelligent systems and the Ph.D. degree in control science and engineering both from the Central South University (CSU), Changsha, China, in 2006 and 2011, respectively.

He is currently an Associate Professor with the School of Information Science and Engineering, CSU. His current research interests include the theory, algorithm design, and interdisciplinary applications of computational intelligence.

Dr. Wang was awarded the Hong Kong Scholar by the Mainland-Hong Kong Joint Postdoctoral Fellows Program, China, in 2013, the Excellent Doctoral Dissertation by Hunan Province, China, in 2013, the New Century Excellent Talents in University by the Ministry of Education, China, in 2013, the 2015 IEEE Computational Intelligence Society Outstanding PhD Dissertation Award, the Hunan Provincial Natural Science Fund for Distinguished Young Scholars, in 2016, the EU Horizon 2020 Marie Sklodowska-Curie Fellowship, in 2016, and a Highly Cited Researcher in computer science by Clarivate Analytics, in 2017. He is currently serving as an associate editor for the *Swarm and Evolutionary Computation*.

Da-Qing Yin received the B.S. degree in automation and M.S. degree in control science and engineering both from Central South University, Changsha, China, in 2014 and 2017. He is working on intelligent search development in Huawei 2012 laboratory.

His current research interests include expensive constrained evolutionary optimization, machine learning, and natural language processing.

Shengxiang Yang (M'00-SM'14) received the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China in 1999.

He is currently a Professor in Computational Intelligence and Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 240 publications.

His current research interests include evolutionary computation, swarm intelligence, computational intelligence in dynamic and uncertain environments, artificial neural networks for scheduling, and relevant real-world applications. He serves as an Associate Editor/Editorial Board Member of seven international journals, such as the *IEEE Transactions on Cybernetics*, *Information Sciences*, *Evolutionary Computation*, and *Soft Computing*.

Guangyong Sun received the B.S. and Ph.D. degrees in Mechanical Engineering from Hunan University, Changsha, China, in 2003 and 2011, respectively. He is currently an ARC Discovery Early Career Researcher with the School of Aerospace, Mechanical and Mechatronic Engineering, Faculty of Engineering, The University of Sydney. His research interests include lightweight design, structural optimization, and automotive safety.

Supplementary File

S-I. Effect of the Parameter Settings

A comprehensive set of experiments were carried out to test the effect of the parameter settings of GLoSADE. All the parameter settings were kept unchanged unless otherwise specified. Again, the 13 test functions from IEEE CEC2006 were utilized for the experimental study. In the experiments, 25 independent runs were implemented for each test function with *MaxFEs*=3000.

1) Effect of the Number of Trial Vectors λ in the Global Surrogate-Assisted Search: In the global surrogate-assisted search, λ trial vectors are generated for each target vector, and then these trial vectors are evaluated by GRNN or the uncertainty-based rule. Table S9 and Fig. S1 in the supplementary file summarize the experimental results in the case of λ alone being changed to 10, 50, 100, and 200.

In the case of λ=10, the mean function error values of test functions $CEC2006_{09}$, $CEC2006_{10}$, $CEC2006_{16}$, $CEC2006_{18}$, and $CEC2006_{19}$ are much worse than those in the case of λ=50 and 100. Also, GLoSADE with λ=200 provides worse mean function error values than GLoSADE with λ=50 and 100 on test functions $CEC2006_{06}$, $CEC2006_{10}$, $CEC2006_{18}$, and $CEC2006_{19}$.

The above discussion suggests that a value between 50 and 100 is suitable for λ.

2) Effect of the Number of the Nearest Points η in the Uncertainty-Based Rule: In the uncertainty-based rule, the uncertainty of a surrogate at a trial vector is computed by making use of the η nearest points in archive *A* to this trial vector. To analyze the effect of η, we changed this parameter to 10, 50, 100, and 200. Table S10 and Fig. S2 in the supplementary file summarize the experimental results.

In the case of λ=10, the algorithm provides the worse mean function error values on test functions $CEC2006_{12}$, $CEC2006_{16}$, and $CEC2006_{18}$. In addition, when using a relatively big value (i.e., 200), the performance degradation occurs on test functions $CEC2006_{06}$, $CEC2006_{07}$, $CEC2006_{10}$, and $CEC2006_{16}$, compared with λ=50 and 100.

Therefore, a value between 50 and 100 is recommended for this parameter.

3) Effect of the Number of the Nearest Points K in the Local Surrogate-Assisted Search: In the local surrogate-assisted search, we made use of the *K* nearest points in archive *A* to build the RBF surrogates for each individual. A small value of *K* will result in insufficient information for building the RBF surrogates. On the contrary, a large value may contain some points distant from the current individual, thus introducing error information. Hence, an appropriate value should be set to *K*. For investigating the effect of *K*, we tested five different *K*: 10, 50, 100, 200, and $K = \max((D+1)(D+2)/2, 100)$. The experimental results are given in Table S11 and Fig. S3 of the supplementary file.

It is obvious from Table S11 that in the case of *K* = 10, 50, 100, and 200, GLoSAD with *K* = 100 exhibits the best overall performance. When *K* = 10, the mean function error values are worse than the other corresponding results on $CEC2006_{01}$, $CEC2006_{09}$, $CEC2006_{10}$, $CEC2006_{12}$, $CEC2006_{16}$, $CEC2006_{18}$, and $CEC2006_{19}$. The mean function error values of test functions $CEC2006_{10}$, $CEC2006_{18}$, and $CEC2006_{19}$ are not good in the case of *K* = 50. When *K* = 200, the algorithm cannot consistently find feasible solutions for test functions $CEC2006_{06}$ and $CEC2006_{07}$. It is necessary to note that if $D < 13$, there is no difference between $K = \max((D+1)(D+2)/2, 100)$ and *K* = 100. Therefore, GLoSADE with *K* = 100 may have performance difference with GLoSADE with $K = \max((D+1)(D+2)/2, 100)$ on test functions $CEC2006_{01}$, $CEC2006_{02}$, and $CEC2006_{19}$. As shown in Table S11, there is no significant performance difference between these two algorithms on $CEC2006_{01}$ and $CEC2006_{02}$. Nevertheless, for $CEC2006_{19}$, GLoSADE with $K = \max((D+1)(D+2)/2, 100)$ is better than GLoSADE with *K* = 100.

The above discussion demonstrates that $K = \max((D+1)(D+2)/2, 100)$ is the best choice for GLoSADE.

Table Captions

- **Table S1** THIRTEEN TEST FUNCTIONS WITH ONLY INEQUALITY CONSTRAINTS FROM IEEE CEC2006. D IS THE NUMBER OF DECISION VARIABLES, ρ IS THE ESTIMATED RATIO BETWEEN THE FEASIBLE REGION AND THE SEARCH SPACE, LI IS THE NUMBER OF LINEAR INEQUALITY CONSTRAINTS, NI IS THE NUMBER OF NONLINEAR INEQUALITY CONSTRAINTS, AND a IS THE NUMBER OF CONSTRAINTS ACTIVE AT THE OPTIMAL SOLUTION.
- **TABLE S2** SIX TEST FUNCTIONS WITH 10 DIMENSIONS (10D) AND 30 DIMENSIONS (30D) INCLUDING ONLY INEQUALITY CONSTRAINTS FROM IEEE CEC2010. D IS THE NUMBER OF DECISION VARIABLES, ρ IS THE ESTIMATED RATIO BETWEEN THE FEASIBLE REGION AND THE SEARCH SPACE, LI IS THE NUMBER OF LINEAR INEQUALITY CONSTRAINTS, AND NI IS THE NUMBER OF NONLINEAR INEQUALITY CONSTRAINTS.
- **Table S3** NINE TEST FUNCTIONS WITH 10 DIMENSIONS (10D) AND 30 DIMENSIONS (30D) INCLUDING ONLY INEQUALITY CONSTRAINTS FROM IEEE CEC2017. D IS THE NUMBER OF DECISION VARIABLES, LI IS THE NUMBER OF LINEAR INEQUALITY CONSTRAINTS, AND NI IS THE NUMBER OF NONLINEAR INEQUALITY CONSTRAINTS.
- **Table S4** EXPERIMENTAL RESULTS OF (μ+μ)-CEP-RBF, FROFI, AND GLOSADE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON SIX TEST FUNCTIONS WITH 10D AND 30D FROM IEEE CEC2010. (#) DENOTES THE NUMBER OF TRIALS IN WHICH AT LEAST ONE FEASIBLE SOLUTION IS FOUND IN THE FINAL POPULATION OVER 25 INDEPENDENT RUNS.
- **Table S5** EXPERIMENTAL RESULTS OF (μ+μ)-CEP-RBF, FROFI, AND GLOSADE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON NINE TEST FUNCTIONS WITH 10D AND 30D FROM IEEE CEC2017. (#) DENOTES THE NUMBER OF TRIALS IN WHICH AT LEAST ONE FEASIBLE SOLUTION IS FOUND IN THE FINAL POPULATION OVER 25 INDEPENDENT RUNS.
- **Table S6** EXPERIMENTAL RESULTS OF GLOSADE_WOU AND GLOSADE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006.
- **Table S7** EXPERIMENTAL RESULTS OF GLOSADE_GLOBAL, GLOSADE_LOCAL, AND GLOSADE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006. (#) DENOTES THE NUMBER OF TRIALS IN WHICH AT LEAST ONE FEASIBLE SOLUTION IS FOUND IN THE FINAL POPULATION OVER 25 INDEPENDENT RUNS.
- **Table S8** EXPERIMENTAL RESULTS OF GLOSADE_RBF, GLOSADE_GP, AND GLOSADE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006.
- **Table S9** EXPERIMENTAL RESULTS OF GLOSADE WITH VARYING λ WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006.

- **Table S10** EXPERIMENTAL RESULTS OF GLOSADE WITH VARYING NUMBER OF THE NEATEST POINTS IN THE UNCERTAINTY-BASED RULE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006.
- **Table S11** EXPERIMENTAL RESULTS OF GLOSADE WITH VARYING K WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006. (#) DENOTES THE NUMBER OF TRIALS IN WHICH AT LEAST ONE FEASIBLE SOLUTION IS FOUND IN THE FINAL POPULATION OVER 25 INDEPENDENT RUNS.

Figure Captions

- **Figure S1** Plot of the mean function error values of GLoSADE with varying λ
- **Figure S2** Plot of the mean function error values of GLoSADE with varying η
- **Figure S3** Plot of the mean function error values of GLoSADE with varying K
- **Figure S4** The shape of 2D NACA0012 airfoil.
- **Figure S5** Evolution of the mean of the best feasible objective function values for the optimal shape design of 2D transonic airfoil by implementing (μ+μ)-CEP-RBF and GLoSADE.

TABLE S1

THIRTEEN TEST FUNCTIONS WITH ONLY INEQUALITY CONSTRAINTS FROM IEEE CEC2006. *D* IS THE NUMBER OF DECISION VARIABLES, ρ IS THE ESTIMATED RATIO BETWEEN THE FEASIBLE REGION AND THE SEARCH SPACE, *LI* IS THE NUMBER OF LINEAR INEQUALITY CONSTRAINTS, *NI* IS THE NUMBER OF NONLINEAR INEQUALITY CONSTRAINTS, AND a IS THE NUMBER OF CONSTRAINTS ACTIVE AT THE OPTIMAL SOLUTION.

Problem	*D*	Type of objective function	ρ	*LI*	*NI*	*a*
$CEC2006_{01}$	13	quadratic	0.0111%	9	0	6
$CEC2006_{02}$	20	nonlinear	99.9971%	0	2	1
$CEC2006_{04}$	5	quadratic	52.1230%	0	6	2
$CEC2006_{06}$	2	cubic	0.0066%	0	2	2
$CEC2006_{07}$	10	quadratic	0.0003%	3	5	6
$CEC2006_{08}$	2	nonlinear	0.8560%	0	2	0
$CEC2006_{09}$	7	polynomial	0.5121%	0	4	2
$CEC2006_{10}$	8	linear	0.0010%	3	3	6
$CEC2006_{12}$	3	quadratic	4.7713%	0	1	0
$CEC2006_{16}$	5	nonlinear	0.0204%	4	34	4
$CEC2006_{18}$	9	quadratic	0.0000%	0	13	6
$CEC2006_{19}$	15	nonlinear	33.4761%	0	5	0
$CEC2006_{24}$	2	linear	79.6556%	0	2	2

TABLE S2

SIX TEST FUNCTIONS WITH 10 DIMENSIONS (10D) AND 30 DIMENSIONS (30D) INCLUDING ONLY INEQUALITY CONSTRAINTS FROM IEEE CEC2010. *D* IS THE NUMBER OF DECISION VARIABLES, ρ IS THE ESTIMATED RATIO BETWEEN THE FEASIBLE REGION AND THE SEARCH SPACE, *LI* IS THE NUMBER OF LINEAR INEQUALITY CONSTRAINTS, AND *NI* IS THE NUMBER OF NONLINEAR INEQUALITY CONSTRAINTS.

Problem	Search space	Type of objective function	ρ		*LI*	*NI*
			10D	30D		
$CEC2010_{01}$	$[0,10]^D$	Non Separable	0.997689	1.000000	1	1
$CEC2010_{07}$	$[-140,140]^D$	Non Separable	0.505123	0.503725	0	1
$CEC2010_{08}$	$[-140,140]^D$	Non Separable	0.379512	0.375278	0	1
$CEC2010_{13}$	$[-500,500]^D$	Separable	0.000000	0.000000	0	3
$CEC2010_{14}$	$[-1000,1000]^D$	Non Separable	0.003112	0.006123	0	3
$CEC2010_{15}$	$[-1000,1000]^D$	Non Separable	0.003210	0.006023	0	3

TABLE S3

NINE TEST FUNCTIONS WITH 10 DIMENSIONS (10D) AND 30 DIMENSIONS (30D) INCLUDING ONLY INEQUALITY CONSTRAINTS FROM IEEE CEC2017. *D* IS THE NUMBER OF DECISION VARIABLES, *LI* IS THE NUMBER OF LINEAR INEQUALITY CONSTRAINTS, AND *NI* IS THE NUMBER OF NONLINEAR INEQUALITY CONSTRAINTS.

Problem	Search space	Type of objective function	*LI*	*NI*
$CEC2017_{01}$	$[-100,100]^D$	Non Separable	0	1
$CEC2017_{02}$	$[-100,100]^D$	Non Separable, Rotated	0	1
$CEC2017_{04}$	$[-10,10]^D$	Separable	0	2
$CEC2017_{05}$	$[-10,10]^D$	Non Separable	0	2
$CEC2017_{13}$	$[-100,100]^D$	Non Separable	2	1
$CEC2017_{19}$	$[-50,50]^D$	Separable	0	2
$CEC2017_{20}$	$[-100,100]^D$	Non Separable	0	2
$CEC2017_{22}$	$[-100,100]^D$	Rotated	2	1
$CEC2017_{28}$	$[-50,50]^D$	Rotated	0	2

TABLE S4
EXPERIMENTAL RESULTS OF (μ+μ)-CEP-RBF, FROFI, AND GLOSADE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON SIX TEST FUNCTIONS WITH 10D AND 30D FROM IEEE CEC2010. (#) DENOTES THE NUMBER OF TRIALS IN WHICH AT LEAST ONE FEASIBLE SOLUTION IS FOUND IN THE FINAL POPULATION OVER 25 INDEPENDENT RUNS.

Problem (10D)	Mean Objective Function Value±Std Dev		
	(μ+μ)-CEP-RBF	FROFI	GLoSADE
$CEC2010_{01}$	-3.21E-01±5.22E-02−	-5.20E-01±3.63E-02−	-6.19E-01±4.43E-02
$CEC2010_{07}$	7.57E+05±8.36E+05−	2.50E+07±1.43E+07−	2.21E+03±2.46E+03
$CEC2010_{08}$	1.97E+05±2.97E+05−	1.61E+08±1.11E+08−	6.56E+04±4.18E+04
$CEC2010_{13}$	-5.68E+01±5.48E+00−	-3.92E+01±3.98E+00−	-6.28E+01±1.13E+00
$CEC2010_{14}$	9.74E+12±1.39E+13−	2.57E+13±1.55E+13−	7.84E+07±1.93E+08
$CEC2010_{15}$	(8)	(4)	(13)
Problem (30D)	Mean Objective Function Value±Std Dev		
	(μ+μ)-CEP-RBF	FROFI	GLoSADE
$CEC2010_{01}$	-3.17E-01±4.77E-02≈	-2.47E-01±1.85E-02−	-3.18E-01±1.95E-02
$CEC2010_{07}$	9.53E+06±8.44E+06−	2.11E+10±7.76E+09−	1.83E+05±1.68E+05
$CEC2010_{08}$	3.68E+09±1.87E+09−	2.50E+10±8.72E+09−	8.11E+06±4.05E+06
$CEC2010_{13}$	-4.21E+01±1.03E+01≈	(0)	-4.20E+01±1.06E+01
$CEC2010_{14}$	1.89E+14±7.57E+13−	1.84E+14±5.52E+13−	8.75E+09±4.58E+09
$CEC2010_{15}$	(15)	(15)	2.71E+12±3.91E+12

"+", "−", and "≈" denote that the performance of the corresponding algorithm is better than, worse than, and similar to that of GLoSADE, respectively.

TABLE S5

EXPERIMENTAL RESULTS OF (μ+μ)-CEP-RBF, FROFI, AND GLOSADE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON NINE TEST FUNCTIONS WITH 10D AND 30D FROM IEEE CEC2017. (#) DENOTES THE NUMBER OF TRIALS IN WHICH AT LEAST ONE FEASIBLE SOLUTION IS FOUND IN THE FINAL POPULATION OVER 25 INDEPENDENT RUNS.

Problem (10D)	Mean Objective Function Value±Std Dev		
	(μ+μ)-CEP-RBF	FROFI	GLoSADE
$CEC2017_{01}$	2.85E+00±3.72E+00＋	5.54E+02±1.62E+02＋	1.08E+03±5.17E+02
$CEC2017_{02}$	4.16E+00±8.99E+00＋	5.08E+02±1.17E+02－	3.60E+02±2.16E+02
$CEC2017_{04}$	1.20E+02±2.41E+01－	1.01E+02±7.91E+00－	8.88E+01±6.16E+00
$CEC2017_{05}$	5.54E+01±5.83E+01－	3.53E+02±1.31E+02－	7.37E+00±6.15E-01
$CEC2017_{13}$	4.89E+03±1.06E+04－	(0)	1.79E+01±3.56E+01
$CEC2017_{19}$	(0)	(0)	(0)
$CEC2017_{20}$	2.35E+00±3.79E-01－	3.24E+00±3.46E-01－	2.19E+00±3.01E-01
$CEC2017_{22}$	2.53E+02±2.21E+02－	(0)	2.12E+01±2.28E+01
$CEC2017_{28}$	(0)	(0)	(0)
Problem (30D)	Mean Objective Function Value±Std Dev		
	(μ+μ)-CEP-RBF	FROFI	GLoSADE
$CEC2017_{01}$	1.91E+04±1.05E+03－	2.77E+04±5.54E+03－	1.24E+04±1.56E+03
$CEC2017_{02}$	1.25E+04±6.37E+03≈	2.65E+04±5.49E+03－	1.13E+04±3.77E+03
$CEC2017_{04}$	3.99E+02±1.53E+01－	5.33E+02±3.86E+01－	3.15E+02±4.05E+01
$CEC2017_{05}$	4.21E+02±1.23E+02－	1.45E+05±7.94E+04－	1.94E+02±1.14E+02
$CEC2017_{13}$	(0)	(0)	(0)
$CEC2017_{19}$	(0)	(0)	(0)
$CEC2017_{20}$	1.13E+01±4.74E-01－	9.76E+00±6.42E-01≈	9.88E+00±3.68E-01
$CEC2017_{22}$	(0)	(0)	(0)
$CEC2017_{28}$	(0)	(0)	(0)

“＋”, “－”, and “≈” denote that the performance of the corresponding algorithm is better than, worse than, and similar to that of GLoSADE, respectively.

TABLE S6
EXPERIMENTAL RESULTS OF GLoSADE_WoU AND GLoSADE WITH 3000 FEs OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006

Problem	Mean Function Error Value±Std Dev	
	GLoSADE_WoU	GLoSADE
$CEC2006_{01}$	4.49E-07±3.22E-07≈	4.73E-07±5.29E-07
$CEC2006_{02}$	4.62E-01±2.27E-02−	4.09E-01±2.74E-02
$CEC2006_{04}$	1.36E-07±5.66E-08≈	2.27E-07±1.10E-07
$CEC2006_{06}$	1.77E-05±5.39E-05−	8.13E-06±1.91E-05
$CEC2006_{07}$	1.10E-07±1.59E-08−	8.13E-08±3.43E-08
$CEC2006_{08}$	4.20E-05±1.30E-04−	8.85E-11±1.45E-11
$CEC2006_{09}$	3.60E-08±4.68E-09+	3.82E-07±6.07E-07
$CEC2006_{10}$	1.48E-05±3.17E-05−	1.71E-06±2.91E-06
$CEC2006_{12}$	2.15E-03±6,27E-03−	3.04E-04±4.54E-04
$CEC2006_{16}$	8.37E-08±1.15E-08≈	8.11E-08±8.85E-09
$CEC2006_{18}$	3.82E-02±8.05E-02−	9.66E-05±3.04E-04
$CEC2006_{19}$	1.49E-03±4.11E-03−	1.03E-05±1.46E-05
$CEC2006_{24}$	3.30E-08±2.44E-08−	1.50E-08±7.74E-09

"+", "−", and "≈" denote that the performance of the corresponding algorithm is better than, worse than, and similar to that of GLoSADE, respectively.

TABLE S7
EXPERIMENTAL RESULTS OF GLOSADE_GLOBAL, GLOSADE_LOCAL, AND GLOSADE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006. (#) DENOTES THE NUMBER OF TRIALS IN WHICH AT LEAST ONE FEASIBLE SOLUTION IS FOUND IN THE FINAL POPULATION OVER 25 INDEPENDENT RUNS.

Problem	Mean Function Error Value±Std Dev		
	GLoSADE_Global	GLoSADE_Local	GLoSADE
$CEC2006_{01}$	1.21E-01±5.64E-02−	4.45E-01±4.72E-01−	4.73E-07±5.29E-07
$CEC2006_{02}$	4.31E-01±4.25E-02−	4.00E-01±2.69E-02≈	4.09E-01±2.74E-02
$CEC2006_{04}$	3.32E+00±1.45E+00−	1.42E+01±6.26E+00−	2.27E-07±1.10E-07
$CEC2006_{06}$	4.14E+00±2.68E+00−	2.35E+02±3.67E+02−	8.13E-06±1.91E-05
$CEC2006_{07}$	2.18E+01±5.38E+00−	3.51E+01±6.97E+01−	8.13E-08±3.43E-08
$CEC2006_{08}$	1.21E-03±2.47E-03−	1.49E-02±2.77E-02−	8.85E-11±1.45E-11
$CEC2006_{09}$	2.78E+00±7.71E-01−	4.03E-06±7.21E-06−	3.82E-07±6.07E-07
$CEC2006_{10}$	(20)	1.68E-06±1.61E-06≈	1.71E-06±2.91E-06
$CEC2006_{12}$	7.29E-03±7.11E-03−	1.13E-03±2.37E-03−	3.04E-04±4.54E-04
$CEC2006_{16}$	3.96E-02±1.10E-02−	1.25E-01±6.91E-02−	8.11E-08±8.85E-09
$CEC2006_{18}$	2.05E-01±3.67E-02−	(13)	9.66E-05±3.04E-04
$CEC2006_{19}$	1.86E+02±5.09E+01−	1.04E+01±6.99E+00−	1.03E-05±1.46E-05
$CEC2006_{24}$	2.72E-03±9.92E-04−	1.14E-07±2.41E-07−	1.50E-08±7.74E-09

"+", "−", and "≈" denote that the performance of the corresponding algorithm is better than, worse than, and similar to that of GLoSADE, respectively.

TABLE S8
EXPERIMENTAL RESULTS OF GLOSADE_RBF, GLOSADE_GP, AND GLOSADE WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006.

Problem	Mean Function Error Value±Std Dev			Runtime (Second)		
	GLoSADE_RBF	GLoSADE_GP	GLoSADE	GLoSADE_RBF	GLoSADE_GP	GLoSADE
$CEC2006_{01}$	4.31E-07±3.88E-07≈	3.58E-08±1.04E-08＋	4.73E-07±5.29E-07	539	7395	271
$CEC2006_{02}$	4.35E-01±2.76E-02－	3.79E-01±2.10E-02＋	4.09E-01±2.74E-02	727	1881	402
$CEC2006_{04}$	4.15E-07±6.85E-07≈	4.77E-07±8.15E-07≈	2.27E-07±1.10E-07	444	2463	218
$CEC2006_{06}$	5.70E-06±6.09E-06≈	6.47E-05±1.34E-04－	8.13E-06±1.91E-05	356	1073	196
$CEC2006_{07}$	1.06E-07±1.93E-08≈	9.39E-08±1.54E-08≈	8.13E-08±3.43E-08	518	4155	265
$CEC2006_{08}$	8.20E-11±2.99E-11≈	8.40E-11±4.45E-12≈	8.85E-11±1.45E-11	425	1275	203
$CEC2006_{09}$	6.51E+00±9.65E+00－	3.23E+01±1.66E+01－	3.82E-07±6.07E-07	550	2609	278
$CEC2006_{10}$	1.19E-06±1.30E-06≈	4.08E-06±4.93E-06－	1.71E-06±2.91E-06	602	3528	317
$CEC2006_{12}$	3.55E-16±4.54E-16＋	8.43E-13±4.87E-13＋	3.04E-04±4.54E-04	407	886	213
$CEC2006_{16}$	1.65E-07±6.69E-08－	1.73E-07±3.02E-08－	8.11E-08±8.85E-09	567	38130	351
$CEC2006_{18}$	5.52E-05±7.06E-05≈	8.27E-07±6.77E-07＋	9.66E-05±3.04E-04	559	6344	297
$CEC2006_{19}$	4.90E-07±3.59E-07＋	7.58E-07±4.55E-07＋	1.03E-05±1.46E-05	606	3237	340
$CEC2006_{24}$	2.86E-08±1.23E-08≈	9.81E-09±3.38E-09≈	1.50E-08±7.74E-09	389	1170	198

“＋”, “－”, and “≈” denote that the performance of the corresponding algorithm is better than, worse than, and similar to that of GLoSADE, respectively.

TABLE S9
EXPERIMENTAL RESULTS OF GLOSADE WITH VARYING λ WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006

Problem	Mean Function Error Value±Std Dev			
	λ=10	λ=50	λ=100	λ=200
$CEC2006_{01}$	2.72E-07±4.13E-08	4.32E-07±4.78E-07	4.73E-07±5.29E-07	5.72E-07±6.68E-07
$CEC2006_{02}$	4.06E-01±5.04E-02	4.10E-01±4.25E-02	4.09E-01±2.74E-02	4.06E-01±1.62E-02
$CEC2006_{04}$	3.55E-07±2.90E-07	2.24E-07±2.09E-07	2.27E-07±1.10E-07	2.22E-07±1.66E-07
$CEC2006_{06}$	3.45E-05±9.07E-05	3.84E-05±1.20E-04	8.13E-06±1.91E-05	6.63E-03±2.09E-02
$CEC2006_{07}$	8.58E-08±2.77E-08	8.77E-08±3.74E-08	8.13E-08±3.43E-08	1.00E-07±2.08E-08
$CEC2006_{08}$	9.38E-11±2.60E-12	2.26E-10±3.28E-10	8.85E-11±1.45E-11	1.32E-10±7.12E-11
$CEC2006_{09}$	6.84E-04±1.45E-03	2.35E-06±5.04E-06	3.82E-07±6.07E-07	1.97E-07±2.43E-07
$CEC2006_{10}$	8.96E-02±2.77E-01	5.36E-04±1.13E-02	1.71E-06±2.91E-06	6.20E-02±1.35E-01
$CEC2006_{12}$	6.41E-04±2.03E-06	5.63E-04±1.78E-03	3.04E-04±4.54E-04	1.74E-08±4.18E-08
$CEC2006_{16}$	5.03E-06±1.54E-05	8.96E-08±6.62E-09	8.11E-08±8.85E-09	9.67E-08±1.22E-08
$CEC2006_{18}$	3.82E-02±8.05E-02	1.92E-04±6.04E-02	9.66E-05±3.04E-04	3.82E-02±8.06E-02
$CEC2006_{19}$	9.21E-01±2.91E+00	8.75E-05±2.40E-04	1.03E-05±1.46E-05	3.69E-04±8.57E-04
$CEC2006_{24}$	2.14E-08±2.22E-08	2.98E-08±1.14E-08	1.50E-08±7.74E-09	3.33E-08±9.98E-09

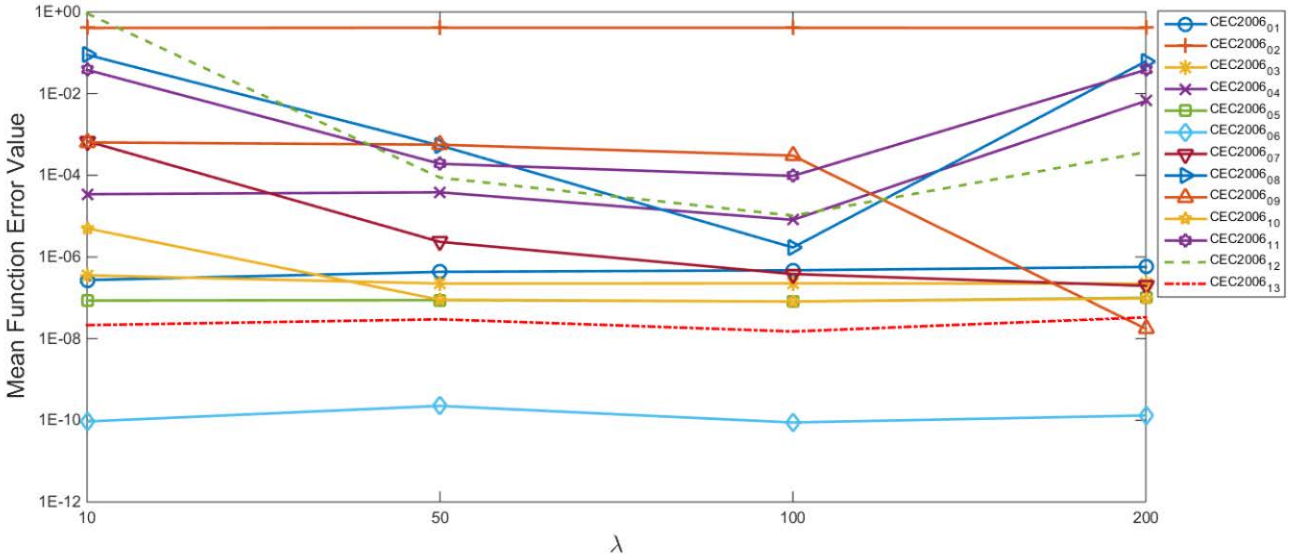

Fig. S1. Plot of the mean function error values of GLoSADE with varying λ

TABLE S10
EXPERIMENTAL RESULTS OF GLOSADE WITH VARYING η WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006.

Problem	Mean Function Error Value±Std Dev			
	$\eta = 10$	$\eta = 50$	$\eta = 100$	$\eta = 200$
$CEC2006_{01}$	3.81E-07±3.77E-07	3.73E-07±3.22E-07	4.73E-07±5.29E-07	3.91E-07±1.73E-07
$CEC2006_{02}$	4.00E-01±4.16E-02	3.85E-01±4.19E-02	4.09E-01±2.74E-02	3.97E-01±3.20E-02
$CEC2006_{04}$	8.50E-07±1.65E-06	3.60E-07±4.27E-07	2.27E-07±1.10E-07	2.74E-07±2.81E-07
$CEC2006_{06}$	1.95E-07±2.59E-07	5.52E-06±1.18E-05	8.13E-06±1.91E-05	6.97E-05±1.73E-04
$CEC2006_{07}$	8.85E-08±3.30E-08	9.53E-08±2.25E-08	8.13E-08±3.43E-08	1.13E-07±1.25E-08
$CEC2006_{08}$	9.56E-11±2.01E-11	1.21E-10±9.39E-11	8.85E-11±1.45E-11	2.78E-10±4.34E-10
$CEC2006_{09}$	1.00E-07±7.75E-08	7.04E-07±1.17E-06	3.82E-07±6.07E-07	6.97E-07±1.01E-06
$CEC2006_{10}$	1.16E+00±3.65E+00	1.53E-03±3.76E-02	1.71E-06±2.91E-06	1.68E+00±5.30E+00
$CEC2006_{12}$	3.64E-03±4.50E-03	5.63E-04±1.78E-03	3.04E-04±4.54E-04	2.39E-11±3.81E-11
$CEC2006_{16}$	2.10E-02±4.42E-02	9.10E-08±5.47E-09	8.11E-08±8.85E-09	7.13E-03±2.25E-02
$CEC2006_{18}$	1.34E-01±9.23E-02	5.73E-02±9.23E-02	9.66E-05±3.04E-04	5.74E-02±9.25E-02
$CEC2006_{19}$	1.09E-05±1.66E-05	1.53E-04±4.62E-04	1.03E-05±1.46E-05	1.13E-05±1.87E-05
$CEC2006_{24}$	4.51E-08±6.24E-08	3.74E-08±1.97E-08	1.50E-08±7.74E-09	4.09E-08±3.06E-08

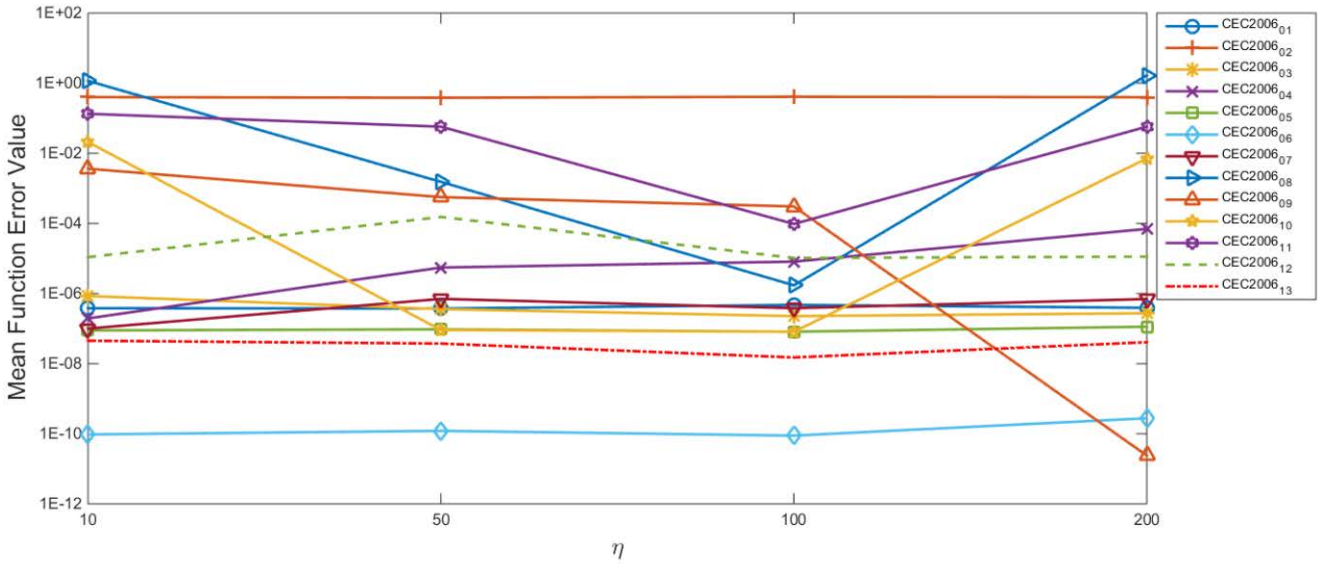

Fig. S2. Plot of the mean function error values of GLoSADE with varying η

TABLE S11
EXPERIMENTAL RESULTS OF GLOSADE WITH VARYING K WITH 3000 FES OVER 25 INDEPENDENT RUNS ON 13 TEST FUNCTIONS FROM IEEE CEC2006. (#) DENOTES THE NUMBER OF TRIALS IN WHICH AT LEAST ONE FEASIBLE SOLUTION IS FOUND IN THE FINAL POPULATION OVER 25 INDEPENDENT RUNS.

Problem	D	Mean Function Error Value±Std Dev				
		K=10	K=50	K=100	K=200	$K = \max((D+1)(D+2)/2, 100)$
$CEC2006_{01}$	13	5.26E-02±2.11E-02	4.80E-07±3.69E-07	4.65E-07±5.19E-07	2.27E-07±8.24E-08	4.73E-07±5.29E-07
$CEC2006_{02}$	20	4.03E-01±3.94E-02	4.32E-01±5.67E-02	4.03E-01±2.34E-02	4.17E-01±3.07E-02	4.09E-01±2.74E-02
$CEC2006_{04}$	5	4.64E-06±8.67E-06	3.35E-07±2.77E-07	2.27E-07±1.10E-07	1.67E-01±5.27E-01	2.27E-07±1.10E-07
$CEC2006_{06}$	2	2.73E-06±4.85E-06	5.16E-05±1.35E-04	8.13E-06±1.91E-05	(20)	8.13E-06±1.91E-05
$CEC2006_{07}$	10	3.17E+00±1.55E+00	1.00E-07±3.29E-08	8.13E-08±3.43E-08	(23)	8.13E-08±3.43E-08
$CEC2006_{08}$	2	7.70E-10±8.05E-10	1.06E-10±3.29E-11	8.85E-11±1.45E-11	4.62E-09±1.40E-08	8.85E-11±1.45E-11
$CEC2006_{09}$	7	1.37E+00±7.85E-01	1.14E-06±1.45E-06	3.82E-07±6.07E-07	9.49E-04±1.25E-03	3.82E-07±6.07E-07
$CEC2006_{10}$	8	2.30E+02±1.59E+02	7.59E-03±1.98E-02	1.71E-06±2.91E-06	1.80E-06±3.69E-06	1.71E-06±2.91E-06
$CEC2006_{12}$	3	3.90E-03±6.89E-03	1.94E-05±6.14E-05	3.04E-04±4.54E-04	7.99E-08±2.53E-07	3.04E-04±4.54E-04
$CEC2006_{16}$	5	1.25E-02±3.58E-02	8.62E-08±1.02E-08	8.11E-08±8.85E-09	7.40E-08±1.98E-08	8.11E-08±8.85E-09
$CEC2006_{18}$	9	1.95E-01±9.64E-02	1.92E-02±6.04E-02	9.66E-05±3.04E-04	2.41E-07±4.19E-07	9.66E-05±3.04E-04
$CEC2006_{19}$	15	1.46E+02±4.76E+01	6.33E-01±1.27E+00	1.90E-03±5.92E-03	6.54E-05±1.90E-04	1.03E-05±1.46E-05
$CEC2006_{24}$	2	1.88E-07±2.63E-07	3.03E-08±1.16E-08	1.50E-08±7.74E-09	1.47E-08±1.25E-08	1.50E-08±7.74E-09

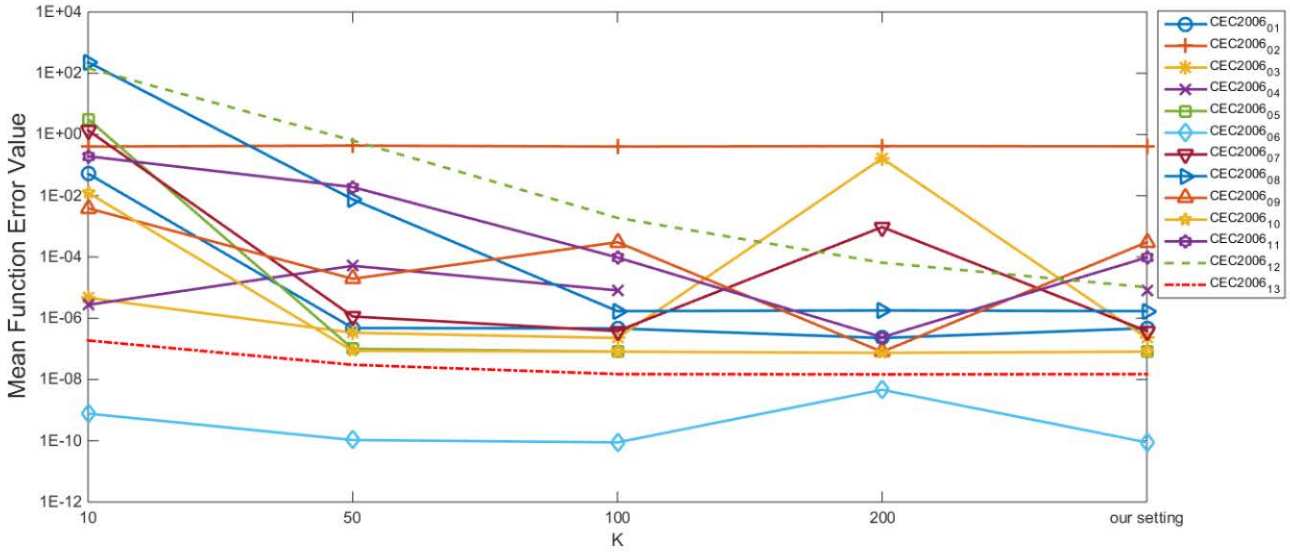

Fig. S3. Plot of the mean function error values of GLoSADE with varying K

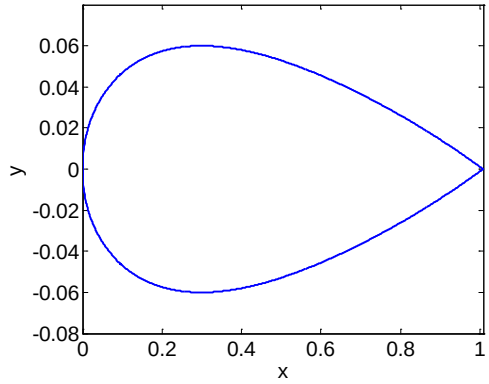

Fig. S4. The shape of 2D NACA0012 airfoil.

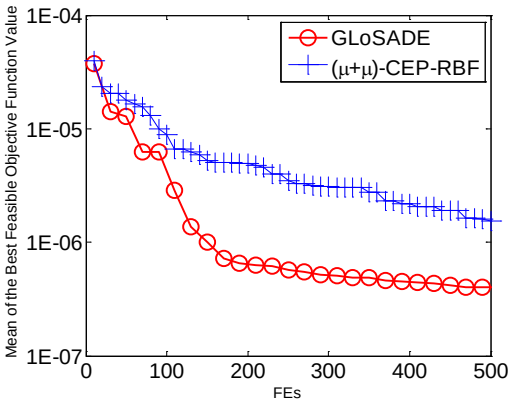

Fig. S5. Evolution of the mean of the best feasible objective function values for the optimal shape design of 2D transonic airfoil by implementing (μ+μ)-CEP-RBF and GLoSADE.