Hybrid surrogate-based constrained optimization with a new constraint-handling method

Yuanping Su, Lihong Xu, *Senior Member, IEEE*, and Erik D. Goodman

***Abstract*—Surrogate-based constrained optimization for some optimization problems involving computationally expensive objective functions and constraints is still a great challenge in the optimization field. Its difficulties are of two primary types. One is how to handle the constraints, especially equality constraints; another is how to sample a good point to improve the prediction of the surrogates in the feasible region. Overcoming these difficulties requires a reliable constraint-handling method and an efficient infill-sampling strategy. To perform inequality- and equality-constrained optimization of expensive black-box systems, this work proposes a hybrid surrogate-based constrained optimization method (HSBCO), and the main innovation is that a new constraint-handling method is proposed to map the feasible region into the origin of the Euclidean subspace. Thus, if the constraint violation of an infeasible solution is large, then it is far from the origin in the Euclidean subspace. Therefore, all constraints of the problem can be transformed into an equivalent equality constraint, and the distance between an infeasible point and the origin in the Euclidean subspace represents the constraint violation of the infeasible solution. Based on the distance, the objective function of the problem can be penalized by a Gaussian penalty function, and the original constrained optimization problem becomes an unconstrained optimization problem. Thus, the feasible solutions of the original minimization problem always have a lower objective function value than any infeasible solution in the penalized objective space. To improve the optimization performance, kriging-based efficient global optimization (EGO) is used to find a locally optimal solution in the first phase of HSBCO, and starting from this locally optimal solution, RBF-model-based global search and local search strategies are introduced to seek global optimal solutions. Such a hybrid optimization strategy can help the optimization process to converge to the global optimal solution within a given maximum number of function evaluations, as demonstrated in the experimental results on 23 test problems. The method is shown to achieve the global optimum more closely and efficiently than other leading methods.**

***Index Terms*—surrogate, constrained optimization, constraint handling, Gaussian penalty function, hybrid optimization**

This work was supported in part by in part by the National Natural Science Foundation of China[grant number 61863015]and [grant number 61973337], and the U.S. National Science Foundation's BEACON Center for the Study of Evolution in Action under Cooperative Agreement DBI-0939454. and the key research and development project of Jiangxi province [grant number 20192ACB60016]

Y. Su, is with Beacon Center, Michigan State University, East Lansing, MI 48824USA, and with Jiangxi University of Science and Technology, Nanchang, 330013China. (e-mail: suyuanping_2003@163.com).

L. Xu, is with Tongji University, Shanghai, 201804 China (e-mail:xulhk@163.com) , Xu is the corresponding author.

E. D. Goodman, is with Beacon Center, Michigan State University, East Lansing, MI 48824 USA(e-mail: goodman@msu.edu).

I. INTRODUCTION

A. Motivation

THE motivation of this work is the need for an effective surrogate-based constrained optimization method to solve the greenhouse climate setpoint problem. The aim of energy-saving greenhouse climate control is not only to create a favorable artificial environment for crop growth, to improve the yield, but also to reduce the energy consumption and production cost as much as possible. However, although several greenhouse control methods have been developed in our previous work [1-5], the practical experience and research results show that the production efficiency of the greenhouse climate control significantly depends on the greenhouse climate setpoint [6-9] in ways not yet captured. To improve the efficiency of the greenhouse climate control, we must optimize the greenhouse climate setpoint by maximizing the crop yield and minimizing energy consumption. However, greenhouse climate setpoint optimization is still a great challenge due to the complex dynamics of the greenhouse system and the great uncertainty of the weather. Since an accurate long-term weather forecast is usually unavailable, the greenhouse climate setpoint obtained by off-line optimization with historical weather data is not reliable enough. Actually, it is also difficult to optimize the greenhouse climate setpoint within a single optimization problem. An effective way is to divide this problem into several multi-objective and single-objective subproblems with different timescales. These subproblems strongly constrain each other, and there are also many inequality and equality constraints in the subproblems. Therefore, we seek to develop effective constrained multi-objective and single-objective optimization methods to solve these subproblems. Since the long-term weather forecast is unreliable, the greenhouse climate setpoint should be solved online using short-term weather forecasts, which requires that the objective and constraints be easy to evaluate. However, simulation of the greenhouse climate and the crop growth is usually computationally expensive, so we propose to use surrogate-based constrained optimization methods to search for the potential globally optimal solution. For this reason, this paper presents a new surrogate-based constrained optimization method, and the aim of this work is to provide a kind of theoretical basis for the follow-up greenhouse climate setpoint optimization study. Besides, the proposed method also can be used to solve many other similar optimization problems, and is independent of the problem that motivated its development.

B. Review of the literature

In many complex engineering optimization problems, the engineering model may be expensive and black-box, so it is usually difficult to find an optimal solution within a given limited computational budget. In this case, traditional optimization methods cannot be used directly to solve the expensive black-box optimization problem. One way to solve these optimization problems is to use a suitable, computationally cheap surrogate to replace the expensive simulation needed to evaluate the objective function. If the number of expensive sampled data is sufficient, a surrogate function constructed using a nonlinear approximation method can accurately approximate the system. Thus, we can use a surrogate-based optimization method to search for a potential global solution within the given computational budget.

The performance of surrogate-assisted optimization algorithms usually depends significantly on the infill method and the accuracy of the surrogate approximations. Therefore, many effective nonlinear approximation methods, such as polynomial response surface (PRS) [10-11], radial basis function (RBF) [12-14], kriging [15-17], support vector machine (SVM) [18-19], multivariate adaptive regression spline (MARS) [20-21], and Gaussian processes [57] are often used to estimate the true objectives and constraints, and based on these surrogate models, many surrogate-based optimization methods have been proposed in recent years [22-30,67-69].

With respect to the infill method, surrogate-based global optimization methods can be usually categorized into three types. One category is probabilistic-improvement-based global optimization. A typical method in this category is the EGO proposed by Jones [31]. In this method, the kriging model is used to estimate the expensive system, and it samples a globally optimal solution candidate by maximizing the expected improvement (EI). Since the kriging model can provide the mean value and variance of the estimation of the system, the infill criterion has strong global exploration and local exploitation ability, and the optimization process can explore sparse regions. In low-dimensional unconstrained cases, EGO can achieve good optimization results, but if there are many design variables, the performance becomes worse, and the computational cost of the kriging rises greatly. Besides EGO, there are also other probability-based optimization methods, such as probability of improvement (PI) [32], generalized EI [33] and augmented EI [34-35].The second category is stochastic-search-based global optimization. The main idea of this kind of method is to generate randomly some sample points in the design domain, and to evaluate the estimations of the objective and constraints through surrogates. Then a best point can be selected from the sampled points to update the surrogate for the next iteration. The good performance of some typical algorithms, such as constrained accelerated random search RBF (CARS-RBF) [36], local metric stochastic RBF (LMSRBF) [37] and mesh adaptive direct search (MADS) [27, 38] has attracted much attention to them. However, a disadvantage of this kind of method is that it usually takes a large number of function evaluations to find a near-optimal solution [39]. The third category is deterministic optimization methods based on surrogates. In this kind of method, since the surrogates are regarded as the analytical approximations of the true objective and constraints function, if the surrogate is accurate enough, many deterministic optimization methods, including gradient-based and derivative-free optimization, can be used to search for a potential global solution. But if the optimal solution is located in a sparse region of the sampled dataset, the optimization process may miss the optimal solution. In this case, it must introduce some sparse sampling methods to improve the surrogates. For example, the sequential approximate optimization (SAO) [40] uses a density function to discover a sparse region of the design variable space. By maximizing the RBF density function, it can sample a point in the largest sparse area.

Generally, most surrogate-based optimization methods can achieve good results for most unconstrained optimization problems, but if the problems are constrained by both equality and inequality constraints, the surrogate-based optimization faces great challenges. If there are many constraints, then the feasible region may become non-convex and non-continuous, even becoming only several isolated points [70]. In this case, it is difficult to find a feasible optimal solution within a given relatively small number of function evaluations. For most surrogate-based constrained optimization methods, the key issue is how to handle the constraints. For kriging-based constrained EGO, Qian presented five constraint-handling methods [41], and Basudhar et al. introduced a support vector machine (SVM) to approximate the constraint boundary [42]. In constrained EGO [43], the EI is penalized by introducing the product of the feasible probabilities of the inequality constraints, but if there are too many inequality constraints, the product of the feasible probabilities may be very small, which makes it difficult for the infill criterion to select a good feasible solution. Therefore, Bagheri modified the total feasible function by introducing a plugin control strategy [44]. Surrogate-based constrained global optimization using space reduction (SCGOSR) uses a weighted penalty method to transform the kriging-based objective and constraints into an augmented function [45]. For RBF-based constrained optimization methods, a common way to search for a feasible solution is to minimize the sum of the constraint violations [46], but using this method makes it easy to neglect small constraint violations, such that it requires a larger number of function evaluations to find feasible solutions. To avoid this disadvantage, Regis proposed a new constraint-handing method [39]. His approach treats each inequality constraint individually, instead of lumping them into one penalty function, and uses the number of the constraint violations to guide the optimization process; however, it cannot handle equality constraints. Actually, a drawback of most reported surrogate-based constrained optimization methods is that they usually cannot handle equality constraints [39, 46]. To solve equality-constrained optimization problems, based on the Self-adjusting constrained optimization by RBF approximation (SACOBRA) [58], Bagheri proposed a method for handling equality constraints for surrogate-based constrained optimization [47]. His method, using relaxing rules, can transform the equality constraints into inequality constraints.

Some constrained optimization methods, such as constrained optimization by radial basis function interpolation in trust regions (CONORBIT) [48] and constrained LMSBRF

(ConstrLMSBRF) [49], usually have good performance, but they require an initial feasible point or multi-start. However, for some complex constrained optimization problems, this requirement is too strict, which greatly limits their application. Therefore, Regis extended the ConstrLMSBRF algorithm [39].The extended ConstrLMSBRF introduces a two-phase structure, and the task of the first phase is to find a feasible solution. When a feasible solution is found, the optimization process switches to the second phase and uses a local metric stochastic response surface method. To improve the performance of surrogate-based constrained optimization, many new optimization strategies have been proposed in recent years—e.g., multi-phase structure [50], multi-start strategy [51-53], multi-surrogate [54], adaptive surrogate-based optimization [55], parallel adaptive sampling optimization [23] and intelligent space exploitation strategy [56]. For some optimization problems, these strategies are effective, but most of them cannot handle equality constraints. Therefore, it is important to develop a new surrogate-based constrained optimization method capable of handling equality constraints.

C. Contribution

The main contribution and innovation of this paper is a novel inequality and equality constraint handling method which can handle both inequality and equality constraints at the same time. This constraint-handling method can transform the equality and inequality constraints into one equivalent equality constraint by mapping the infeasible design domain into a hypercubic Euclidean subspace. The origin of the hypercubic Euclidean subspace represents the feasible region. Therefore, if a point in the hypercubic Euclidean subspace is far from the origin, it has a large constraint violation. The advantage of this constraint-handling method is that no matter how many constraints there are, they can be integrated into a single new equality constraint. But based on the proposed constraint handling method, a constrained optimization problem can be transformed into an unconstrained optimization problem by introducing a penalty term, such that we can use some effective classical optimization methods to solve the new unconstrained optimization problem.

Since multi-surrogates may greatly help to find a better global solution due to their special approximation features, this work uses both kriging and RBF to construct the surrogates of the objective and constraint functions, and the proposed algorithm also introduces two optimization mechanisms-—kriging-based EGO and RBF-based optimization. Therefore, the proposed algorithm has two phases. In the first phase, kriging-based EGO is introduced to find a locally optimal solution, then starting from this solution, RBF-based optimization is used to search for the globally optimal solution.

The rest of this paper is organized as follow. Section II introduces the kriging and RBF modeling methods. In Section III, the novel constraint handling method is presented. In Section IV, we discuss the details of the proposed surrogate-assisted optimization method. Section V describes the experimental design and results, and presents comparisons with other methods. Section VI presents conclusions.

II. Surrogate Models for Expensive Black-box Functions

Generally, a constrained single optimization problem can be expressed as

$$\begin{aligned} &\min f(\mathbf{x}) \\ &s.t.\, h_i(\mathbf{x}) = 0, i = 1,\cdots l, \\ &g_i(\mathbf{x}) \le 0, i = l+1,\cdots,n, \\ &lb \le \mathbf{x} \le ub \end{aligned} \tag{1}$$

where $\mathbf{x} = [x_1,\cdots,x_d]$ is the design variable vector, $f(\mathbf{x})$ is the objective function, and $h_i(\mathbf{x})$ and $g_i(\mathbf{x})$ represent the equality and inequality constraints, respectively. lb and ub are the lower and upper bounds of the design variables.

For many engineering problems, the objective function $f(\mathbf{x})$ and the constraint functions $h_i(\mathbf{x})$ and $g_i(\mathbf{x})$ are not explicitly known and/or are computationally expensive. In this case, we can use an RBF interpolation model and/or a kriging model to approximate them.

Suppose we have a training dataset $\mathbb{D} = \{X = [\mathbf{x}_1,\cdots,\mathbf{x}_N], Y = [y_1,\cdots,y_N]\}$, where $\mathbf{x}_i = [x_{i,1},\cdots,x_{i,d}]$ is a d-dimensional design vector, and if we use a kriging model to estimate the true objective function; it then can be expressed as

$$f(\mathbf{x}) = \mu(\mathbf{x}) + z(\mathbf{x}) \tag{2}$$

where $\mu(x)$ is a constant global model, and $z(\mathbf{x}) \sim N(0,\sigma^2)$ is a Gaussian random variable with zero mean and variance σ^2. Then the posterior estimation of $\mu(\mathbf{x})$ and the posterior variance of $z(\mathbf{x})$ can be evaluated as

$$\bar{\mu}(\mathbf{x}) = \frac{\mathbf{1}^T\mathbf{R}^{-1}Y}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}} \tag{3}$$

$$\hat{\sigma}^2 = \frac{(Y-\mathbf{1}\mu(\mathbf{x}))\mathbf{R}^{-1}(Y-\mathbf{1}\mu(\mathbf{x}))}{N} \tag{4}$$

Thus, the best linear unbiased predictor of $f(\mathbf{x})$ can be evaluated by

$$\hat{f}_{kriging}(\mathbf{x}) = \bar{\mu}(\mathbf{x}) + \mathbf{r}(\mathbf{x}^*,\mathbf{x})^T\mathbf{R}^{-1}(Y-\mathbf{1}\bar{\mu}(\mathbf{x})) \tag{5}$$

where $\mathbf{r}(\mathbf{x}^*,\mathbf{x}) = [r^1(\mathbf{x}^*,\mathbf{x}),\cdots,r^N(\mathbf{x}^*,\mathbf{x})]$ is the linear vector of correlation between the unknown point $\mathbf{x}$ to be predicted and the known sample points $\mathbf{x}^*$, and $\mathbf{R}$ denotes the $N\times N$ correction matrix with entries (i,j) denoting $Corr(\varepsilon(\mathbf{x}_i),\varepsilon(\mathbf{x}_j))$. The mean squared error is

$$s^2(\mathbf{x}) = \hat{\sigma}^2\left[\mathbf{1}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1-\mathbf{1}\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}^TR^{-1}\mathbf{1}}\right] \tag{6}$$

If we use the RBF model with a linear polynomial to approximate the true objective function, it then can be expressed by

$$\begin{aligned} &\hat{f}_{RBF}(\mathbf{x}) = \sum_{i=1}^{N}\lambda_i\psi\left(\left\|\mathbf{x}-\mathbf{x}_i\right\|\right) + p(\mathbf{x}) \\ &with\ p(\mathbf{x}) = c_1 + c_2x_1 + \cdots + c_{d+1}x_d \end{aligned} \tag{7}$$

where $||\cdot||$ is Euclidean norm, $p(\mathbf{x})$ is a linear polynomial, $\psi(\cdot)$ is the basis function, and λ_i is the coefficient of the i-th basis function. It should be noted that the basis function can have several alternative forms, such as cubic, $\psi(l) = l^3$, thin plate spline, $\psi(l) = l^2 \log l$, Gaussian, $\psi(l) = \exp(-(l/\gamma)^2)$, and multiquadric, $\psi(l) = \sqrt{l^2 + \gamma^2}$, where γ is a parameter. Since the cubic basis function is simpler than the others, and does not require additional parameters, this work uses the cubic basis function to construct the RBF model.

Define $\mathbf{Q} \in R^{N\times N}$ as the Euclidean distance matrix of the training data set; its elements then can be evaluated by $q_{ij} = \psi\left(\left\|\mathbf{x}_i - \mathbf{x}_j\right\|\right)$, $i, j = 1, \cdots, N$. Also define a matrix $\mathbf{H} \in R^{N\times(d+1)}$ with the i-th row being $[1, \mathbf{x}_i]$, and denote $\lambda = [\lambda_1, \cdots, \lambda_N]$ and $c = [c_1, \cdots, c_{d+1}]$; then we have

$$\begin{bmatrix} \mathbf{Q} & \mathbf{H} \\ \mathbf{H}^T & \mathbf{0}_{(d+1)\times(d+1)} \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ c \end{bmatrix} = \begin{bmatrix} Y \\ \mathbf{0}_{(d+1)} \end{bmatrix} \tag{8}$$

If the rank of matrix $\mathbf{H}$ is $d+1$, then Eq. (8) can be solved analytically, and the parameters λ and c have unique solutions. This condition requires that the points in the training data set be mutually independent. In the same way, all the equality and inequality constraint functions are approximated by the RBF model in this work; we then have $h_i(\mathbf{x}) \approx \hat{h}_{i_{RBF}}(\mathbf{x})$ and $g_i(\mathbf{x}) \approx \hat{g}_{i_{RBF}}(\mathbf{x})$.

III. CONSTRAINT HANDLING

An important issue of the constrained optimization is how to handle the equality and inequality constraints. This work proposes a new method to handle the equality and inequality constraints of the optimization problems.

Assume that the equality and inequality constraints $h_i(\mathbf{x})$ and $g_i(\mathbf{x})$ are analytical; then the infeasible region of the design space can be mapped into an infeasible Euclidean subspace, and the feasible region can be mapped to a point, as shown in Fig. 1.

Suppose we have a design point $\mathbf{x}$ in the design space, then the Euclidean distance r_i from this point to the region that satisfies the i-th constraint condition can be defined as $r_i = C_i(\mathbf{x})$.

For the equality constraint $h_i(\mathbf{x})$, the mapping function $C_i(\mathbf{x})$ can be designed as follows:

$$C_i(\mathbf{x}) = \begin{cases} \left|h_i(\mathbf{x})\right| & if\, h_i(\mathbf{x}) \neq 0 \\ 0 & if\, h_i(\mathbf{x}) = 0 \end{cases} \tag{9}$$

For the inequality constraint $g_i(\mathbf{x})$, $C_i(\mathbf{x})$ can be designed as follows:

$$C_i(\mathbf{x}) = \begin{cases} g_i(\mathbf{x}) & if\, g_i(\mathbf{x}) > 0 \\ 0 & otherwise \end{cases} \tag{10}$$

Then, the union Euclidean distance from the design point $\mathbf{x}$ to the feasible region r can be evaluated as

$$r = \sqrt{r_1^2 + \cdots + r_n^2} \tag{11}$$

Obviously, if r is zero, then it means that the design point $\mathbf{x}$ is a feasible solution, and $r = 0$ is the unique feasible point in the Euclidean subspace. If the equality and inequality constraints are known, then we can find their maximum violation $C_i^{\max}$ by solving the following optimization problem.

$$C_i^{\max} = \max C_i(\mathbf{x}) \tag{12}$$

But an issue is that problem (12) cannot be directly solved, and the true maximum violation $C_i^{\max}$ may be unknown, if the constraints are expensive and black-box. In this case, it must select a reference constraint violation $\tilde{C}_i^{\max}$ to normalize the Euclidean distance. A simplest way is to select the maximum constraint violation in the training data set as $\tilde{C}_i^{\max}$. However, the constraint violation C_i may be larger than $\tilde{C}_i^{\max}$. Therefore, to keep the normalized Euclidean distance within [0, 1], if $C_i > \tilde{C}_i^{\max}$, the constraint violation must be modified according to the following rule.

$$\hat{C}_i = \zeta(C_i) = \begin{cases} \tilde{C}_i^{\max} & if\, C_i > \tilde{C}_i^{\max} \\ C_i & otherwise \end{cases} \tag{13}$$

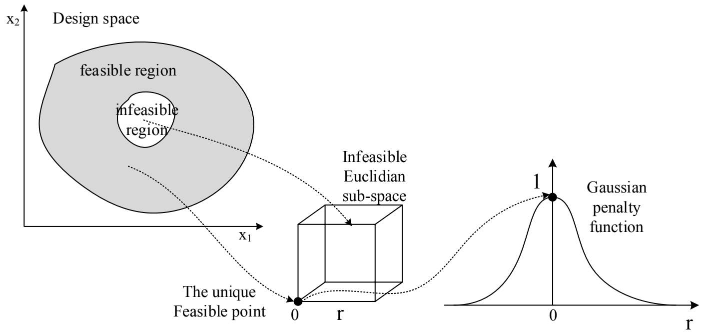


Fig. 1. Infeasible region in the design space mapped to a hypercube in the Euclidean space

Thus, the Euclidean distance r_i can be normalized as

$$\hat{r}_i = \hat{C}_i \Big/ \tilde{C}_i^{\max} \tag{14}$$

Due to modification (13), the normalized Euclidean distance $\hat{r}_i$ is always not bigger than one, and it also results in

$$\hat{r} = \sqrt{\hat{r}_1^2 + \cdots + \hat{r}_n^2} \tag{15}$$

These equations mean that the infeasible region can be mapped as a hypercube in the Euclidean subspace. Then, the constrained optimization problem (1) becomes an equality-constrained optimization problem— i.e., it results in

$$\begin{aligned} &\min f(\mathbf{x}), \\ &s.t.\ \hat{r}(\mathbf{x}) = 0 \end{aligned} \tag{16}$$

We can see that no matter how many equality and inequality constraints there are, using this constraint handling method can transform the constraints into a single equality constraint. If the constraint functions are expensive and black-box, we can use the estimations $\hat{h}_{i_{RBF}}(\mathbf{x})$ and $\hat{g}_{i_{RBF}}(\mathbf{x})$ obtained from their RBF models to evaluate the constraint violation $\hat{r}(\mathbf{x})$.

It should be pointed out that if the constraint functions are known explicitly, we can use the union Euclidean distance to judge the feasibility of a sampling point in the design space, but if the constraints are unknown, and have to be estimated by surrogates, then the condition $\hat{r} = 0$ does not necessarily mean the sampling point is feasible, due to the uncertainty of the surrogates. Therefore, sometimes even when $\hat{r} = 0$, it may still sample an infeasible point, but in most cases, this infeasible point is near the feasible boundary. Likewise, the condition $\hat{r} > 0$ also does not necessarily mean the sampling point is infeasible. Due to the approximation error of the surrogates, feasible points near the feasible boundary may be misjudged, and if the global optimal solution is near the feasible boundary, we may not find it using the proposed method. A way to avoid such a case is to set a small feasible threshold $r_{threshold}$. Then, if the union Euclidean distance of a sampling point is smaller than $r_{threshold}$, then this point can be regarded as a feasible point. However, how to select the feasible threshold is an important issue. For some simple problems, even if we set a small fixed threshold, the proposed method may still achieve good performance, but if there are many constraints, the accumulative estimation error of the union Euclidian distance may cause the disappearance of the estimated feasible region. In this case, the feasible threshold should be adjusted adaptively. In Section III of the supplementary material, we present a way to adjust this parameter.

IV. Hybrid Surrogate-based Constrained Optimization

A. Framework of the algorithm

The proposed hybrid surrogate-based constrained global optimization procedure includes two phases. The first phase uses kriging-based efficient global optimization to find a feasible local solution. Its aim is to find a locally optimal feasible solution for the RBF-based optimization in the second phase. It uses the amount of equality constraint violation to penalize the infill criterion to help the optimization process to converge to a feasible solution. In low-dimensional cases, kriging-based EGO usually can find the globally optimal solution if a suitable number of iterations is allowed. However, if the dimension of the design space is high, the advantage of the kriging-based EGO may disappear, and the optimization process may fall into a local optimum, and have difficulty escaping from it, such that the best solution found may also not be further improved in subsequent iterations. To deal with such cases, we define a counter n_{KriRBF} to count the number of consecutive iterations at which the current best solution is *not* improved. If this count grows beyond a given threshold N_{KriRBF}, then the algorithm switches to its second phase——RBF-model-based optimization.

The RBF-model-based optimization includes two sub-phases: one is global search, and the other is local search. Since the proposed constraint-handling method can transform a constrained optimization problem into an unconstrained optimization problem, the global search may directly find the globally optimal solution in most cases. But if the feasible region is not convex, or only includes some isolated points, then the global search may not always find a feasible solution. Therefore, if the global search returns an infeasible candidate solution, then it is better to discard this solution and to search for a feasible candidate solution by minimizing the equivalent equality constraint, to replace the infeasible solution. If the global search cannot further improve the current best solution in a series of consecutive iterations, then the optimization process goes to the local search sub-phase. The aim of local search is to find a better solution in the neighborhood of the current best solution. We define another counter n_{RBF} to count the number of consecutive iterations in which the current best solution is not improved. If the counter exceeds a given threshold N_{RBF}, then the switch between global search and local search is executed. The whole procedure of the hybrid surrogate-based constrained optimization is illustrated in Fig. 2.

B. Kriging-based constrained efficient global optimization

Kriging-based constrained optimization is used to find a locally optimal solution in the first phase. In an unconstrained case, we can sample a point by maximizing the expected improvement acquisition function $EI(\mathbf{x})$, i.e., $\mathbf{x}^{opt} = \arg\max_{\mathbf{x}\in D} EI(\mathbf{x})$. This acquisition function measures the expected amount by which observing $f(\mathbf{x})$ leads to improvement over the current minimum $y_{\min}$, and can be evaluated by

$$EI(\mathbf{x}) = (y_{\min} - \bar{\mu}(\mathbf{x}))\Phi\left(\frac{y_{\min} - \bar{\mu}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{y_{\min} - \bar{\mu}(\mathbf{x})}{s(\mathbf{x})}\right) \tag{17}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative distribution function and probability density function, respectively. Since this infill criterion can trade off global search and local search, it can generally ensure global convergence. Therefore, this method has been used in some engineering optimization problems [59-60].

However, if there are constraint conditions, the difficulty of solving the optimization problem will rise greatly. The crucial issue is how to handle the constraints. If the constraints

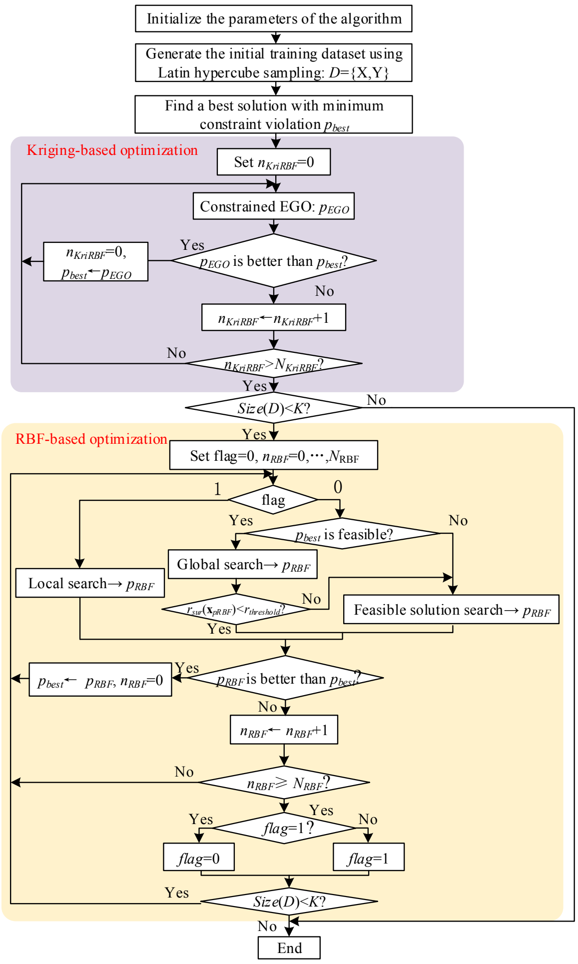


Fig. 2. Framework of HSBCO, where r_{surr} is the constraint violation estimated by the surrogates of the constraints, and $r_{threshold}$ is the threshold of the estimated constraint violation, "*flag*" is a label used for the switch between the local search and the global search in the second phase

are also expensive and black-box, and are estimated by kriging models, then we can use the Gaussian union cumulative probability to penalize the infill criterion in the efficient global optimization methods [43]; i.e., we can obtain a modified expected improvement acquisition function by multiplying by a penalty factor $P(\mathbf{x})$. Then the modified infill criterion can be described as

$$EI_{\text{mod}}(\mathbf{x}) = P(\mathbf{x}) \cdot EI(\mathbf{x}) \tag{18}$$

where $P(\mathbf{x}) = \prod_{i=1}^{n} P(g_i(\mathbf{x}) \le 0)$, and $P(g_i(\mathbf{x}) \le 0)$ is the probability that $g_i(\mathbf{x})$ is feasible. Obviously, the probability of the constraint is available only for inequality constraints. If there are equality constraints, it is difficult to evaluate their probabilities. Therefore, this method is available only for cases having only inequality constraints.

In addition, a great disadvantage of this approach is that the union probability may make the infill criterion very small in most cases, such that the optimization process can easily fall into a local optimum, and the feasible solution found may be far from the true optimal solution. To circumvent this disadvantage, Bagheri introduced a plugin control mechanism to modify the expected improvement function [44]. His results indicate that for some test problems, this method can achieve good optimization performance, and the feasibility rate can reach 99%, while in some cases, few feasible solutions can be found. It is clear that the constraint handling is still the main barrier to efficient global optimization.

The proposed constraint handling method transforms the constraints into equality constraint $\hat{r}(\mathbf{x}) = 0$. Thus, if expensive black-box constraints are estimated by RBF models, then we can define a new constraint penalty function as

$$P(\mathbf{x}) = \exp\left(-\frac{\hat{r}^2}{2\rho^2}\right) \tag{19}$$

where the parameter ρ is a small constant set by the user. It is clear that this penalty function is a Gaussian function, which has good selection ability. If $\hat{r} = 0$, the penalty is removed, as shown in Fig. 1. However, the modified infill criterion cannot always ensure the solution to be feasible even if the expected improvement is optimal. For example, suppose we have a feasible solution $\mathbf{x}_a$ and an infeasible solution $\mathbf{x}_b$; it is possible that $EI_{\text{mod}}(\mathbf{x}_a) = EI(\mathbf{x}_b)$. In this case, the optimization process may converge to the infeasible solution $\mathbf{x}_b$, and the constraint optimization may not be solved. A way to circumvent this issue is to reduce the parameter ρ. If ρ is small, the constraint penalty function has strong selection ability. Therefore, the sampled solutions are almost feasible. Nevertheless, setting ρ as a small constant is not a good idea, because as stated by Runarsson [61-62], the infeasible solutions in the population can often be very helpful for approaching convergence to a feasible solution, especially at the earlier stages of the optimization process. Therefore, we allow this parameter to be adjusted adaptively in the optimization process. But how do we determine whether the optimization process has converged to a local infeasible optimum? Actually, there is an interesting phenomenon—i.e., after the optimization process falls into a local infeasible optimum, the sampled points of consecutive iterations are very close together. That is to say, we can use the minimum Euclidean distance $d_{\min}$ between the current sampled point and the points in the population to decide whether the optimization process has fallen into a local infeasible optimum. If this situation occurs, we can reduce the parameter ρ to increase the selection ability of the penalty function, such that the optimization process may escape from the local infeasible optimum. The minimum Euclidean distance can be obtained by

$$d_{\min} = \min\left(\left\|\mathbf{x} - \mathbf{x}_i\right\|, \mathbf{x}_i \in X\right) \tag{20}$$

We can set a threshold ε for $d_{\min}$ to adjust the parameter ρ. If the condition $d_{\min} < \varepsilon$ is satisfied, then the parameter ρ will be updated by $\rho \leftarrow \alpha \cdot \rho$, where α is a small constant, $0 < \alpha < 1$.

The kriging-based constrained efficient global optimization algorithm can be found in Subsection A of Section I of the supplementary material.

C. RBF-based constrained global optimization

Generally, if there are a large number of design variables, the kriging model may lose the efficiency and become more complex and computationally expensive. Therefore it may be difficult for kriging-based efficient global optimization to achieve good optimization performance in high-dimensional cases, but an RBF model might achieve better approximation performance in such cases. Therefore, RBF-based constrained optimization is employed to search the global solution in the second phase. The proposed RBF-based constrained optimization includes two parts—global and local search—and starts from a local optimal point obtained by the kriging-based optimization. In most cases, this starting point is feasible or is close to the feasible region.

1) Global Search

If the expensive black-box objective function can be approximated by the RBF model, then the optimization problem can be rewritten as follows:

$$\begin{aligned} &\mathbf{x}_{GlobOpt} = \arg \min_{\mathbf{x}\in[lb,ub]} \hat{f}_{RBF}(\mathbf{x}) \\ &s.t.\ \hat{r}(\mathbf{x}) = 0 \end{aligned} \tag{21}$$

and the constraint violation $\hat{r}(\mathbf{x})$ is also evaluated by the RBF models of the constraints. In contrast to the original optimization problem, the optimization problem shown in (21) is easier to solve. If the starting point is feasible, and the RBF model is accurate enough, then it may find a globally optimal solution using gradient-based constrained optimization algorithms. But these methods may not always ensure global convergence, and sometimes the algorithm may find an infeasible optimal solution. In this case, we need to minimize the constraint violation to find a new feasible solution candidate for the next iteration; i.e., we have to solve the following optimization problem:

$$\mathbf{x}_{feasible} = \arg \min_{\mathbf{x}\in[lb,ub]} \hat{r}(\mathbf{x}) \tag{22}$$

However, if the points in the training dataset are excessively close, then it may cause the rank of matrix $\mathbf{H}$ to be smaller than $d+1$. Thus, the parameter matrix of the left side of Eq. (8) may be singular, and the obtained coefficients λ and c may not be accurate. In this case, the RBF model cannot ensure the accuracy of the prediction, and the optimization process may converge to an infeasible point. To avoid this phenomenon, like COBRA [39], the algorithm must introduce a distance constraint for optimization problems (21) and (22), i.e., the optimization problems can be rewritten, respectively, as follows:

$$\begin{aligned} &\mathbf{x}_{GlobOpt} = \arg \min_{\mathbf{x}\in[lb,ub]} \hat{f}_{RBF}(\mathbf{x}) \\ &s.t.\ \hat{r}(\mathbf{x}) = 0, \\ &\left\|\mathbf{x} - \mathbf{x}_i\right\| \geq \eta_k, i = 1,\cdots,N \end{aligned} \tag{23}$$

$$\begin{aligned} &\mathbf{x}_{feasible} = \arg \min_{\mathbf{x}\in[lb,ub]} \hat{r}(\mathbf{x}) \\ &s.t. \left\|\mathbf{x} - \mathbf{x}_i\right\| \geq \eta_k, i = 1,\cdots,N \end{aligned} \tag{24}$$

where the distance threshold η_k can be adjusted with the iteration. For the k-th iteration, the distance threshold can be determined by $\eta_k = \sigma_k \cdot \ell[lb,ub]$, where $\ell[lb,ub]$ is the length of the smallest side of the box $[lb,ub] \in R^{d\times d}$, and $\sigma_k \in \Theta = [\theta_1,\cdots,\theta_m]$. We can first set a fixed parameter set Θ before starting the optimization process, and σ_k can be determined in the following way:

$$\sigma_k = \begin{cases} \theta_j & if\ j = \mathrm{mod}(k,m) \neq 0 \\ \theta_m & if\ j = \mathrm{mod}(k,m) = 0 \end{cases} \tag{25}$$

where k is the iteration index.

For the equality-constrained optimization problem (21), the common solvers, such as the interior-point algorithm and particle swarm optimization algorithm (PSO) in the Matlab optimization toolbox, usually have difficulty converging to the globally optimal solution. If we use the penalty function (19) to penalize the approximated objective function, then problem (21) becomes a general optimization problem without equality constraints, which can be described as follows:

$$\begin{aligned} &\mathbf{x}_{GlobOpt} = \arg \min_{\mathbf{x}\in[lb,ub]} \hat{f}_{RBF}(\mathbf{x}) + (1 - P(\mathbf{x}))\cdot\Gamma \\ &s.t.\ \left\|\mathbf{x} - \mathbf{x}_i\right\| \geq \eta_k, i = 1,\cdots,N \end{aligned} \tag{26}$$

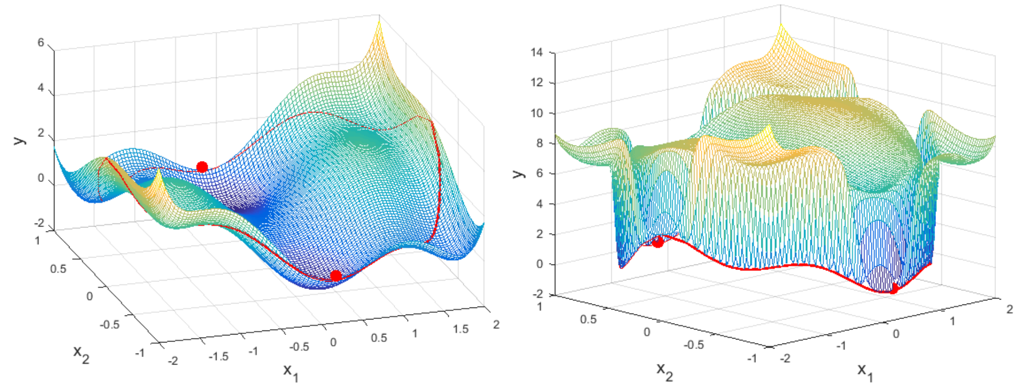
where Γ is a given large constant, and can be determined from the maximum and minimum objective values $f_Y^{\max}$ and $f_Y^{\min}$ in the current training dataset $\mathbb{D}$; i.e., $\Gamma = f_Y^{\max} - f_Y^{\min}$. Obviously, only if $P(\mathbf{x})$ is one, the optimization process can converge to the global feasible region. In contrast to problem (23), if we need not consider the inequality constraint $\left\|\mathbf{x} - \mathbf{x}_i\right\| \geq \eta_k$, problem (26) becomes an unconstrained optimization problem, so it is easier to solve. The penalty item in problem (26) may make the infeasible region have much larger objective values than the feasible region. For example, we can use the following SYP test problem to illustrate this feature:

$$\begin{aligned} &\min y(x_1,x_2) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \\ &s.t.\ \frac{x_1^2}{4} + x_2^2 - 1 = 0, with -2 \leq x_1 \leq 2, -1 \leq x_2 \leq 1 \end{aligned} \tag{27}$$

This problem has an equality constraint and two symmetrical globally optimal solutions, namely $f(-0.1818, 0.9959) = f(0.1818, -0.9959) = -0.0839$. The original objective surface is as shown in Fig. 3(a), and if the constant Γ is set as 10, and the variance of the Gaussian penalty function is set as 0.1, then the penalized objective surface is as shown in Fig. 3(b). From Fig. 3, it can be seen that the feasible region becomes deep gullies, and the infeasible region becomes high mountains. If the variance of the Gaussian penalty function is set as a large number, the surface of the penalized objective may be smooth. From Fig. 3, it can be observed that after the objective is transformed, the feasible global minima become the global minima of the whole objective space.

From the global search algorithm, it can be seen that the aim of the global search is to find a better feasible solution. Since it cannot ensure that the solution obtained by solving problem (26) is feasible, we have to carry out the feasible solution search (24) to find a feasible solution candidate to improve the sampling efficiency.

2) Local Search



(a) (b)

Fig. 3. The original objective surface (a) and the penalized objective surface (b). The red solid line is the feasible region, and the large red points are the globally optimal solutions.

If the surrogates are accurate enough, the global search can find a globally optimal solution in most cases, but if the globally optimal solution is located at a sharp peak, the surrogate of the objective may not capture it. In this case, the optimal solution found by the global search may deviate from the globally optimal solution, and cannot be further improved in consecutive iterations. However, it may be helpful if the optimization process switches to a local search routine.

A key issue for the local search is how to determine the search neighborhood. Generally, we can first initialize the radius of the neighborhood R as $R = \upsilon \cdot (ub - lb)$, where $0 < \upsilon < 1$. If the current best solution is far from the boundary of the design space, then such a radius for the neighborhood is suitable. But if the current best solution is close to a boundary; then such a radius may cause part of the neighborhood to be outside the design space. In this case, we have to modify the lower and upper bounds of the design variables to make the neighborhood remain in the design space. Therefore, for design variable x_i, the lower and upper bounds for local search can be determined respectively as $LB_i = \max[x_{best,i} - R_i, lb_i]$ and $UB_i = \min[x_{best,i} + R_i, ub_i]$.

If the feasible region is non-convex, then the neighborhood may include some infeasible solutions. Therefore, for a given radius R, local search may converge to a certain infeasible optimal solution. In this case, we can reduce the radius adjustment parameter by $\upsilon \leftarrow \upsilon / \beta$, where $\beta > 1$, and repeat the local search until finding a feasible better solution or that radius R is less than given threshold τ.

The second issue is how to find a better solution. Generally, the neighborhood is small, so like the method used in ConstrLMRBF [49], it can generate randomly many solution candidates. In some cases, some design variable values of the current best solution match those of the globally optimal solution. Therefore, we want to retain some design variable values from that current best solution in the random solutions. To do so, the algorithm can select only some of the design variables to change at random, according to a given selection probability p_{select}. For example, let it first generate randomly a solution $\mathbf{z} = [z_1, \cdots, z_d]$ within [0, 1]. Then, if $z_i < p_i$, $i = 1, \cdots, d$, the design variable $x_{best,i}$ is selected to change in the neighborhood, and if $z_i \geq p_i$, then it sets $z_i = 0$. Thus, a new random solution can be determined by $\mathbf{x}_{rand} = \mathbf{x}_{best} + R \odot \mathbf{z}$, where $\odot$ is the scalar product operator. But if solution $\mathbf{x}_{rand}$ is outside the neighborhood, it must be corrected to keep it in the neighborhood, i.e., if $x_{rand,i} < LB_i$, then set $x_{rand,i} = LB_i$, and if $x_{rand,i} > UB_i$, then set $x_{rand,i} = UB_i$.

The local search may find a better solution than the current best solution, but a possible situation should be also considered—i.e., if the best solution is not improved after several consecutive iterations, the algorithm must switch to the global search routine. We can define a threshold N_{RBF} for the number of consecutive iterations in which the best solution is not improved. Thus, if the global search or local search cannot find a better solution after several consecutive iterations, the RBF-based optimization is switched between global searching and local searching, as shown in Fig. 2. The procedure of the RBF-based optimization can be found in Subsection B of Section I of the supplementary material.

Now we can give the whole process of the proposed algorithm, and the pseudo code is described as follow:

Algorithm1: Hybrid surrogate based constrained optimization

Input: Number of design variables d, lower and upper bounds of design variables lb and ub.

Output: Approximation of globally optimal solution $p_{opt}(\mathbf{x}_{opt}, y_{opt})$

1. Set the maximum number of iterations K
2. Set initial Latin hypercube sample size: $N_{lhs} = 11(d+1)$
3. Perform Latin hypercube sample: $X = lhsdesign(d, N_{lhs})$
4. Correct the sample points: $X = lb + (ub - lb) \cdot X$
5. Evaluate the objective: $Y = Objective(X)$
6. Evaluate the constraints: $C = Constraint(X)$

```
Find the maximum constraint C̃^max = [C̃_1^max, ⋯, C̃_{n_constr}^max] in C.
Evaluate the constraint violation: R = ConstraintViolation(X, C̃^max)
Obtain training dataset: 𝔻 = {X, Y}
Find the best solution p_best(x_best, y_best) with the best constraint violation r_best ∈ R
Set the threshold N_KriRBF for the switch between kriging-based optimization and RBF-based optimization, and the threshold N_RBF for the switch between global search and local search in the RBF-based optimization.
The first phase: kriging-based optimization
[x_best, y_best, r_best, 𝔻] = KrigingOptimization(𝔻, p_best, r_best, C̃^max, lb, ub, K, N_KriRBF)
If size(𝔻) < K
    The second phase: RBF-based optimization
    [x_opt, y_opt, r_opt] = RBFOptimization(𝔻, p_best, r_best, lb, ub, K, N_RBF)
    return p_opt(x_opt, y_opt)
Else
    return p_opt(x_opt, y_opt) = p_best(x_best, y_best)
End If
```

where *lhsdesign* is the program routine of Latin hypercube sampling, and *Objective* and *Constraint* are the program routines of the objective and constraint evaluations, respectively, *ConstraintViolation* is the program routine of the constraint violation evaluation.

V. Numerical Experiments

A. Alternative optimization methods and test problems

The proposed HSBCO algorithm is compared with three alternative surrogate-based constrained optimization methods to evaluate its optimization performance. The three methods are RBF-based constrained global optimization (RCGO), developed by Wu [46]; and constrained optimization by radial basis function interpolation (COBRA) and Extended ConstrLMSRBF, developed by Regis [39]. Essentially, RCGO and COBRA have similar flowcharts. Their main idea is to find a feasible solution by minimizing the constraint violation in the first phase and to find a better candidate solution by minimizing the RBF-approximated objective function in the second phase. Their performances greatly depend on the efficiency of the solver "*fmincon*" in the Matlab toolbox. In contrast, Extended ConstrLMSRBF does not depend on the *fmincon* solver, but uses local mesh search to find a better solution in the neighborhood of the current best solution. The reason that we use these three methods to validate the proposed method is that they have good performance as reported by the authors, although they cannot handle equality constraints. Since we could not obtain the authors' original source codes, we have implemented their Matlab versions following the flowcharts presented in the papers.

This work considers 23 constrained test problems. For inequality-constrained problems, the proposed method is compared with RCGO, COBRA and Extended ConstrLMSRBF. For equality-constrained problems, it is compared with the constraint handling (EH) method proposed by Bagheri [47], but we use COBRA to search for the optimal candidate point instead of SACOBRA, because both algorithms can achieve similar performance on the G-problems; thus, we call that method COBRA-EH rather than SACOBRA-EH in this paper. The properties of the test problems considered are described in Subsection A of Section II of the supplementary material. More information about G1~G24 can be found in reference [63], SE, TSD and PVD refer to [45], and RB (Rosenbrock function) and MB (Mishra's Bird function) refer to [64] and [65].

B. Experiment

In this work, all the algorithms are performed in the Matlab environment, and each algorithm is run independently 20 times on each test instance. The number of iterations (function evaluations) after initial sampling is set as 107 for all test instances. The parameters used in RCGO, COBRA and Extended ConstrLMSRBF algorithms are set the same as reported in the literature. The following is the special parameter settings of HSBCO:

(1) Kriging-based constrained EGO routine
  a) Initial variance of Gaussian penalty function: $\rho = 0.1$
  b) Minimum Euclidean distance: $\varepsilon = 0.001$
  c) The threshold $r_{threshold} = 0.0001$
  d) Maximum number N_{KriRBF} of the consecutive iterations in which the best solution is not improved: the parameter is set as 50 for 2-dimensional test instances, and is set as 20 for other test instances.
  e) Initial value of the adaptive parameter α : $\alpha = 0.1$
  f) This work uses particle swarm optimization method to search the optimal solution, and the "*particleswarm*" optimizer in the Matlab optimization toolbox is used to maximize the expected improvement, and the population size is set as 100; the maximum iteration number is set as 100.
  g) The DACE toolbox [66] is used to build the kriging model, and zero-order basis function is selected.

(2) RBF-based constrained optimization routine
  a) Maximum number N_{RBF} of consecutive iterations in which the best solution is not improved before a switch is triggered: $N_{RBF} = 5$
  b) The threshold $r_{threshold} = 0.0001$
  c) Variance of the Gaussian penalty function: $\rho = 0.1$ for all inequality-constrained cases except for G24 and PVD, and $\rho = 0.01$ for all equality-constrained cases and G24 and PVD.
  d) Initial flag for global and local search: $flag = 0$
  e) Parameter Θ is set as $\Theta = [0.1, 0.05, 0.01, 0.005, 0.001, 0.0005]$
  f) Initial value and minimum value of parameter υ : $\upsilon_{ini} = 0.2, \upsilon_{\min} = 0.001$
  g) Adjustment parameter β is set to 2
  h) Number of random solutions m in local search: $m = \min[1000(d+1), 20000]$

i) Selection probability is set as $p_{select} = 0.6$

j) The "*fmincon*" solver in the Matlab optimization toolbox is used as the optimizer for global search, and the "*interior-point*" algorithm is selected for local search. It should be pointed out that other operators such as genetic algorithms or active set algorithms could also be used to search for the optimal solution.

C. Results and Discussion

In this experiment, we fixed the number of the function evaluations after initial sampling to compare fairly the performances of the four considered methods, because COBRA, RCGO and Extended ConstrLMSRBF must first find a feasible solution in the first phase. If they cannot find any feasible solution in the first phase, the second phase will not be carried out to further find the globally optimal solution. Therefore, they require finding a good starting point before the algorithms COBRA, RCGO and Extended ConstrLMSRBF are started. But in this experiment, we did not give any prior starting point, and all starting points of the 20 trials for all algorithms were selected from the random initial training dataset, so the starting points may be infeasible in most cases. Thus, the best solutions found during the earlier iterations are infeasible, and sometimes there are not any feasible solutions to be found within 107 iterations, such as when RCGO was applied to G6 and G10, and Extended ConstrLMSRBF to G10. Tables 1 and 2 illustrate the best, median, worst and average objective values of the globally optimal solutions found, and also present the standard deviations of the 20 trials. To better document the performances of the algorithms, the supplementary material also presents figures which illustrate the known best solutions, the median and average progress of the objective values and the constraint violations of the 20 trials.

For most inequality-constrained test problems, the four algorithms achieved good results within the allowed number of function evaluations, and the best solutions found are very close to the known best solutions. In some cases, they found new best solutions—e.g., all algorithms found a new best solution of G8. The median solutions of the problems found by HSBCO are also very close to the best solutions found, implying that the proposed algorithm has good uniform stability. But the performance of HSBCO in converging rapidly to the feasible region may not always be better than COBRA, RCGO and Extended ConstrLMSRBF. In fact, HSBCO needs more function evaluations to find a global optimum on some problem instances; the progress can be found in the supplementary material. For problem G1, COBRA and RCGO can quickly find a feasible solution in the 20 trials, while HSBCO cannot find any feasible solution in the earlier stage of the optimization process. Even then, HSBCO did better than Extended ConstrLMSRBF, because in some trials, Extended ConstrLMSRBF did not ever find any feasible solution within the allowed number of function evaluations. On G1, RCGO achieved the best result, and it also found the known best solution. However, these methods do not always work well, e.g., for G2, none of the methods found the known best solution. The best solution among the four algorithms was found by COBRA, but the median of RCGO was better than COBRA's. Although the median of Extended ConstrLMSRBF was the worst, its uniform stability was best, and from Table 1, it can be seen that Extended ConstrLMSRBF had the minimum STD, and the STD of HSBCO was smaller than those of COBRA and RCGO. For some problems that are easy to solve, all the algorithms achieved similar performances—e.g., for G4, G8, TSD and MB, the STDs of the algorithms were all small, and the best solutions found were very close to the known best. From Table 1, one can see that HSBCO usually has smaller STD than others, but in some cases, it may not converge to the known best—e.g., for G9, the global optimal solution found by HSBCO was far from the known best, and was worse than found by other algorithms. The results of G2 and G9 indicate that HSBCO did not always work better on all the problems than the other algorithms, but HSBCO did always find a good feasible solution within the allowed maximum iterations. For inequality-constrained optimization problems, it can be seen from the figures in Subsection B of Section II of the supplementary material and from Table 1 that the four methods each have their own special advantages. Therefore, it is difficult to conclude which method is best. But the results illustrated in Table 1 indicate that HSBCO achieved better performance in most cases.

It should be noted that the true globally optimal solution of G12 is unknown, because the globally optimal solution depends on the parameters a_1, a_2, a_3 of the constraint. Different parameters may lead to quite different globally optimal solutions. The details on G12 can be found in Subsection A of Section II of the supplementary material. In this experiment, the three parameters were set as 1, 2 and 3, respectively. In this case, the true global optimum is unknown; but from Table 1, one can see that the four algorithms could find similar feasible optimal solutions, and their uniform stabilities on this instance were also good. Nevertheless, the best solution found by HSBCO was better than those found by the other algorithms, while the convergence of RCGO was best among them.

In contrast to COBRA, RCGO and Extended ConstrLMSRBF, another advantage of HSBCO is that it can solve equality-constrained optimization problems. In fact, most surrogate-based constrained optimization methods are only able to deal with inequality constraints. SACOBRA-EH transforms equality-constrained optimization problems into inequality constrained optimization problems by relaxing the equality constraints. To verify the performance of HSBCO in comparison with COBRA-EH, the results on eight test problems are shown in Table 2, and the progress of the best objective values and constraint violations are illustrated in the supplementary material.

From Table 2 and the supplementary material, it can be seen that HSBCO worked well on the eight equality-constrained test problems, and found new best solutions on G5, G11, G13, G15 and G23, while COBRA_EH worked worse than HSBCO on most instances, and sometimes COBRA_EH could not find any feasible solutions, such as on G5 and G15. But on test problem G3, COBRA_EH worked better than HSBCO. On G21, neither HSBCO nor COBRA_EH could ensure that the optimization process converged to the known best solution, which implies that there is room for further improvement of both methods. The results on the eight equality-constrained problems considered indicate that although the problems may be constrained by many equality constraints, HSBCO can generally find the global optimum with a given maximum number of function evaluations, and has good uniform stability and convergence.

For better comparison, we put the medians of the global optima of the test problems found by the algorithms in one figure, i.e., Fig. 24S of the supplementary material.

Table 1. Objective values of solutions found in 20 trials of 107 iterations each, for the 15 inequality test problems considered.

Test problem	Algorithms	Best	Median	Worst	Mean	STD
G1	HSBCO	**-14.99998**	**-14.98854**	**-13.80837**	**-14.81792**	**4.134E-01**
	COBRA	-14.985	-13.81286	-11.46901	-13.57856	1.04816E+00
	RCGO	-15	-15	-13.82812	-14.88281	3.6069E-01
	Extended ConstrLMSRBF	-14.99884	-14.99480	-12.03531	-14.36595	9.5245E-01
G2	HSBCO	-0.49627	-0.31242	-0.25713	-0.33996	7.560E-02
	COBRA	-0.67959	-0.34380	-0.18689	-0.38939	1.5272E-01
	RCGO	-0.58309	-0.41095	-0.21163	-0.41913	1.2474E-01
	Extended ConstrLMSRBF	-0.47106	-0.29238	-0.24806	-0.32032	6.866E-02
G4	HSBCO	-30665.733	-30664.34	-30649.591	-30662.447	4.80998E+00
	COBRA	-30628.302	-30618.583	-30596.412	-30617.976	8.812203E+00
	RCGO	-30665.539	-30662.145	-30582.494	-30650.580	24.01205E+00
	Extended ConstrLMSRBF	-30665.492	-30665.424	-30664.503	-30665.367	2.1079E-01
G6	HSBCO	-6962.309	-6961.873	-6557.036	-6916.418	1.11433E+02
	COBRA	-6920.984	-6782.986	-6782.826	-6806.335	4.127084E+01
	RCGO	*-7118.608*	*532.341*	*9953.045*	*738.332*	*3.17853E+03*
	Extended ConstrLMSRBF	-6958.552	-6946.6	-6923.166	-6945.942	8.04258E+00
G7	HSBCO	**24.3062**	**24.3127**	**25.0136**	**24.4914**	3.069E-01
	COBRA	24.7261	25.4567	25.4569	25.4202	1.633E-01
	RCGO	25.0046	25.0052	3964.0486	1201.1014	1.841E+03
	Extended ConstrLMSRBF	25.4492	750.0372	3955.4115	1200.7544	1.407E+03
G8	HSBCO	-0.10545	-0.10460	-0.02535	-0.09277	2.748E-02
	COBRA	-0.10545	-0.07289	-0.03155	-0.06894	3.746 E-02
	RCGO	-0.10546	-0.03158	1.259008	-0.05803	4.543 E-02
	Extended ConstrLMSRBF	-0.10546	-0.03158	-0.03158	-0.05374	3.473 E-02
G9	HSBCO	724.1145	962.5547	1991.2574	966.9856	2.8499E+02
	COBRA	716.2124	922.7852	1625.0275	979.7986	2.4384E+02
	RCGO	700.2203	937.7656	1420.8775	939.7307	1.9391E+02
	Extended ConstrLMSRBF	681.7241	702.9737	929.1332	721.5283	5.7917E+01
G10	HSBCO	**7063.071**	**7095.382**	**7158.917**	**7096.206**	**2.5386E+01**
	COBRA	7186.534	20532.093	29953.148	19840.582	6.5569E+03
	RCGO	*7400.745*	*16845.971*	*24085.180*	*15823.118*	*4.5987E+03*
	Extended ConstrLMSRBF	*2100*	*5107.771*	*13902.672*	*6296.189*	*4.3703E+03*
G12	HSBCO	-0.62139	-0.62138	-0.62137	*-0.65348*	8.40E-03
	COBRA	-0.61643	-0.61244	-0.60766	-0.61242	1.94E-03
	RCGO	-0.62139	-0.62137	-0.62130	-0.62136	2E-05
	Extended ConstrLMSRBF	-0.62139	-0.62138	-0.62132	-0.62137	2E-05
G24	HSBCO	-5.50801	-5.50801	-5.50801	-5.50801	5.757504E-09
	COBRA	-5.50391	-5.49301	-4.40498	-5.27819	4.4367E-01
	RCGO	-5.50801	-4.79994	-4.1668	-5.01986	4.4987E-01
	Extended ConstrLMSRBF	-5.50799	-5.50786	-4.41986	-5.39906	3.3487E-01
SE	HSBCO	-1.17427	-1.17427	-1.17427	-1.17427	1.9927E-08
	COBRA	-1.1508	-1.14295	7.89984	0.26766	2.999E+00
	RCGO	-1.17427	-1.17395	4.19725	0.70588	2.6285E+00
	Extended ConstrLMSRBF	-1.17427	-1.17425	7.73355	0.22897	2.9777E+00
TSD	HSBCO	0.01263	0.01270	0.01435	0.01285	4.5E-04
	COBRA	0.01284	0.01290	0.01900	0.01353	1.57E-03
	RCGO	0.01267	0.01268	0.01282	0.01271	5E-05
	Extended ConstrLMSRBF	0.01267	0.01272	0.015885	0.01304	8.4E-04
PVD	HSBCO	5885.332	5885.3333	5927.6434	5889.8620	1.1975E+01
	COBRA	6115.414	6173.320	6718.420	6225.616	1.5388E+01
	RCGO	5885.332	5885.333	5908.709	5886.501	5.2270E+00
	Extended ConstrLMSRBF	5944.754	6363.596	7249.425	6402.542	3.4590E+02
MB	HSBCO	-106.7645	-106.7627	-106.7454	-106.7607	4.85E+02
	COBRA	-106.7645	-106.7643	-106.764	-106.7643	1.4E-04
	RCGO	-106.7645	-106.7641	-21.5171	-102.5016	1.9062E+01
	Extended ConstrLMSRBF	-106.7645	-106.7645	-106.7645	-106.7645	4E-07
RB	HSBCO	4.2870E-10	**3.0979E-05**	**0.01124**	**0.00062**	**2.50E-02**
	COBRA	1.00867	1.00886	1.00895	1.00886	6.374E-05
	RCGO	2.3835e-11	0.99889	4.17366	1.10769	7.5534E-01
	Extended ConstrLMSRBF	0.99889	0.99892	0.99905	0.99893	4.0165E+00

Note: The results written in bold are the solutions found by the proposed method that are better than the best solutions found by other algorithms
The results written in italic are the infeasible solutions found by the algorithms

Table 2. Objective value of solutions found in 20 trials, each with 107 iterations, on eight equality test problems

Test problem	Algorithms	Best	Median	Worst	Mean	STD
G3	HSBCO	-0.99939	-0.99571	-0.91843	-0.98203	2.493E-01
	COBRA-EH	-1	-0.99997	-0.99983	-0.99996	3.5777E-05
G5	HSBCO	*5110.039*	*5110.583*	*5125.558*	*5112.266*	*3.6933E+00*
	COBRA-EH	*3819.83*	*4166.878*	*6058.95*	*4368.067*	*5.7648E+02*
G11	HSBCO	0.74969	0.75010	0.75190	0.75024	4.4E-04
	COBRA-EH	0.75001	0.75144	0.86122	0.76259	2.695E-02
G13	HSBCO	0.00228	0.36659	1.553099	0.589129	5.772E-01
	COBRA-EH	0.00787	0.03906	1.0000	0.18008	3.246E-01
G15	HSBCO	961.396	961.617	961.707	961.583	1.1174E-01
	COBRA-EH	961.715	961.841	964.119	962.467	8.2437E-01
G21	HSBCO	358.455	359.191	368.974	360.245	2.6704E+00
	COBRA-EH	411.6701	538.1523	984.1346	593.3961	1.7630E+02
G23	HSBCO	-401.842	-400.263	-400.260	-400.341	3.534E-01
	COBRA-EH	-164.7593	-62.8336	191.8542	-35.8663	1.3299E+02
SYP	HSBCO	-0.08752	-0.08431	-0.08204	-0.08478	1.646E-03
	COBRA-EH	0.17209	0.172099	2.10650	0.36554	5.9539E-01

Note: The results written in bold are the solutions found by the proposed method that are better than the best solutions found by other algorithms
The results written in italic are the infeasible solutions found by the algorithms

From the results on the 23 test problems, we can conclude that the proposed constraint-handling method is effective, and no matter how many inequality and equality constraints are present, HSBCO can achieve good optimization performance in most trials for most of the constrained problems.

The average evaluation numbers for the optimum found by different methods in 20 trials are listed in Table S2 of the supplementary material.

It should be pointed out that although the globally optimal solutions of the test problems found by the four algorithms are very close to the known best, some of them are not completely feasible solutions, and slightly violate the constraints. Therefore, strictly, such solutions are infeasible solutions, so some globally optimal solutions found by HSBCO may have smaller objective values than the known best solutions or the solutions found by the other algorithms, such as on G6 and TSD. In this case, it is difficult to say that the solutions found by HSBCO are better than the known best solutions or those found by the other algorithms. For better observation, we provide the constraint violations of the best solutions found by HSBCO in the supplementary material.

VI. Conclusion

For most surrogate-based constrained optimization problems, there are three important issues to address. One is how to accurately approximate the true objective and constraints using the surrogates. Generally, different kinds of surrogates may perform differently. If the training data is not sufficient, the approximate results may be quite different. One way to improve the performance of surrogate modeling is to use several different kinds of surrogates to estimate the objective and the constraints. Therefore, this work uses both kriging and RBF models to approximate the objective and constraints. Another reason to introduce both models is that the efficiencies and optimization performances of the kriging-based and RBF-based optimization methods may be different, so using such a hybrid surrogate-based optimization method may achieve better results. The second and key issue is how to handle the constraints, especially the equality constraints. Most constrained optimization methods can only solve inequality constrained problems, but the proposed HSBCO can handle both inequality and equality constraints at the same time by mapping the feasible region to be the origin of a Euclidean subspace. Thus, all the constraints are finally transformed into a single equality constraint, and the Gaussian penalty function can transform the equivalent equality-constrained optimization problem into an unconstrained optimization problem. In the new objective space, the feasible region always has lower (better) values than the infeasible region. Therefore, unconstrained optimization methods can be directly used to solve the equivalent unconstrained optimization problem. The transformation of the problem to this more tractable form is also the main contribution of this work. However, model errors in the surrogates of the constraints are inevitable, so the estimated constraint violation may be not accurate. In this case, the feasibility of solutions near to the feasible boundary may be misclassified, and the optimization process may not converge to the global optimal point. Therefore, the third key issue is how to reduce the misclassification of feasible sampled points, which is very important for surrogate-based constrained optimization. One way is to set a feasibility threshold for the estimated constraint violation to define a robust domain, but how to set the feasibility threshold is also an important issue. Although an adaptive method is proposed to adjust the threshold in this work, and that method is shown to be useful, it does not represent a complete solution to the problem. Therefore, this issue should be studied intensively in future work. Nevertheless, the results on 23 test problems verify the good optimization performance of HSBCO, and also indicate the effectiveness of the proposed hybrid optimization strategy.

References

[1]. Y. Su, L. Xu, and E. D. Goodman, "Nearly dynamic programming NN-approximation–based optimal control for greenhouse climate: A simulation study," Optimal Control Application and Methods, vol.39, pp.638–662. 2018.

[2]. Y. Su, L. Xu, and E. D. Goodman, "Control Allocation-based Adaptive Control for Greenhouse Climate," International Journal of System and Sciences, vol.49, no.6, pp.1146-1163, 2018.

[3]. Y. Su, L. Xu, and E. D. Goodman, "Greenhouse Climate Fuzzy Adaptive Control Considering Energy Saving," International Journal of Control,

Automation and Systems, vol.15, no.4, pp. 1936-1948, 2017.
[4]. Y. Su, L. Xu, "Towards discrete time model for greenhouse climate control," Engineering in Agriculture, Environment and Food, vol. 10, no.2, pp.157-170, 2017.
[5]. Y. Su, L. Xu, and D. Li, "Adaptive Fuzzy Control of a Class of MIMO Nonlinear System With Actuator Saturation for Greenhouse Climate Control Problem," IEEE Transactions on Automation Science and Engineering, vol.13, no.2, pp.1-17, 2016.
[6]. P. Jones, J. W. Jones, and Y. Hwang, "Simulation For Determining Greenhouse Temperature Setpoints," Transactions of the ASAE, vol.33, no.5, pp.1722-1728, 1990.
[7]. I. Seginer, G. Shina, LD. Albright, and LS. Marsh, "Optimal temperature setpoints for greenhouse lettuce," Journal of Agricultural Engineering Research, vol.49, no.3, pp.209-226, 1991.
[8]. M. Tchamitchian, R. M. Clouaire, and J. Lagier, et al., "SERRISTE: A daily set point determination software for glasshouse tomato production," Computers and Electronics in Agriculture, vol.50, pp.25- 47, 2006.
[9]. EJ. Van Henten and J. Bontsema, "Time‐scale decomposition of an optimal control problem in greenhouse climate management," Control Engineering Practice, vol.17, pp.88‐96, 2009.
[10]. GE. Box and NR. Draper, "Empirical model-building and response surfaces," John Wiley & Sons, New York, 1987
[11]. RH. Myers and CM. Anderson-Cook, "Response surface methodology: process and product optimization using designed experiments," John Wiley & Sons, New York, 2009.
[12]. Q. Zhou, P. Jiang, and X. Shao et al., "A variable fidelity information fusion method based on radial basis function," Advanced Engineering Informatics, vol. 32, pp.26–39, 2017.
[13]. MJ. Powell, "The theory of radial basis function approximation," Advanced Numerical Analysis, vol. 2, pp.105–210, 1992.
[14]. K. Amouzgar and N. Stromberg, "Radial basis functions as surrogate models with a priori bias in comparison with a posteriori bias," Structural multidisciplinary Optimization. vol. 55, no.4, pp.1453–1469, 2017.
[15]. JPC. Kleijnen, "Kriging metamodeling in simulation: a review," European Journal of Operational Research, vol. 192, no.3, pp.707–716, 2009.
[16]. J. Sacks, WJ. Welch, and TJ. Mitchell et al., "Design and analysis of computer experiments," Statistical. Science, vol. 4, no.4, pp.409–423, 1989.
[17]. K. Kerry and KA. Hawick, "Kriging interpolation on high-performance computers," In: High-Performance computing and networking. Springer Berlin Heidelberg, New York, USA, pp 429–438, 1998.
[18]. V. Vapnik, SE. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," Proceeding NIPS'96 Proceedings of the 9th International Conference on Neural Information Processing Systems, pp. 281–287, Denver, Colorado, December 03, 1996.
[19]. SR. Gunn, "Support vector machines for classification and regression," ISIS technical report, May, 14, 1998.
[20]. JH. Friedman, "Multivariate adaptive regression splines," The annals of statistics, vol. 19, no.1, pp.1-67, 1991.
[21]. JH. Friedman, "Fast MARS," Stanford University. Dept. of Statistics. Laboratory for Computational Statistics, 1993. https://statistics.stanford.edu/research/fast-mars
[22]. Y. Wu, Q. Yin, and H. Jie et al., "A RBF-based constrained global optimization algorithm for problems with computationally expensive objective and constraints," Structural and Multidisciplinary Optimization, vol. 58, pp.1633–1655, 2018.
[23]. RT. Haftka, D. Villanueva, and A.Chaudhuri, "Parallel surrogate-assisted global optimization with expensive functions - a survey," Structural and Multidisciplinary Optimization, vol. 54, no.1, pp.3–13, 2016.
[24]. L. M. Rios, and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," Journal of Global Optimization, vol. 56, no.3, pp. 1247–1293, 2013.
[25]. S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," Structural and Multidisciplinary Optimization, vol. 41, pp. 219–241, 2010.
[26]. K. Khac, V. Claudia, D. Ambrosio, Y. Hamadi, and L. Liberti, "Surrogate‐based methods for black‐box optimization," International transactions in operational research, vol.24, no.3, pp.393-424, 2017.
[27]. C. Audet, and J. E. Dennis, "Mesh Adaptive Direct Search Algorithms for Constrained Optimization," SIAM Journal on Optimization, vol. 17, no. 2, pp. 188–217, 2006.
[28]. G. Liuzzi, S. Lucidi, and M. Sciandrone, "Sequential Penalty Derivative-Free Methods for Nonlinear Constrained Optimization," SIAM Journal on Optimization, vol. 20, no.5, pp. 2614–2635, 2010.
[29]. K. Holmström, "An Adaptive Radial Basis Algorithm (ARBF) for Expensive Black-Box Global Optimization," Journal of Global Optimization, vol. 41, no.3, pp. 447–464, 2008.
[30]. D. Huang, T. T. Allen, W. I. Notz, and N. Zeng, "Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models," Journal of Global Optimization, vol. 34, no. 3, pp. 441–466, 2006.
[31]. DR. Jones, M. Schonlau, and WJ. Welch, "Efficient global optimization of expensive black-box functions," Journal of Global Optimization, vol. 13, no.4, pp.455–492, 1998.
[32]. DR. Jones. "A taxonomy of global optimization methods based on response surfaces," Journal of Global Optimization, vol. 21, no.4, pp.345–383, 2001.
[33]. MJ. Sasena, P. Papalambros, and P. Goovaerts, "Exploration of metamodeling sampling criteria for constrained global optimization," Engineering Optimization, vol. 34, no.3, pp.263–278, 2002.
[34]. D. Huang, T.T. Allen, and W. I. Notz et al., "Global optimization of stochastic black-box systems via sequential kriging meta-models," Journal of Global Optimization, vol. 34, no.3, pp.441–466, 2006a.
[35]. D. Huang, T.T. Allen, W.I. Notz et al. "Sequential kriging optimization using multiple-fidelity evaluations," Structural and Multidisciplinary Optimization, vol. 32, no.5, pp.369–382, 2006b.
[36]. L. Nuñez, R. G. Regis, and K. Varela, "Accelerated Random Search for constrained global optimization assisted by Radial Basis Function surrogates," Journal of Computational and Applied Mathematics, vol. 340, pp.276–295, 2018.
[37]. 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.
[38]. M.A. Abramson and C. Audet, "Convergence of mesh adaptive direct search to second-order stationary points," SIAM Journal on Optimization, vol.17, no.2, pp.606–19, 2006.
[39]. RG. 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.
[40]. S. Kitayama, M. Arakawa, and K. Yamazaki, "Sequential Approximate Optimization using Radial Basis Function network for engineering optimization," Optimization & Engineering., vol.12, pp.535–557, 2011.
[41]. J. Qian, J. Yi, Y. Cheng, J. Liu, and Q. Zhou, "A sequential constraints updating approach for Kriging surrogate model‐assisted engineering optimization design problem," Engineering with Computers, vol.36, pp. 993–1009, 2020.
[42]. A. Basudhar, C. Dribusch, and S. Lacaze et al., "Constrained efficient global optimization with support vector machines," Structural multidisciplinary Optimization, vol.46, no.2, pp.201–221, 2012.
[43]. N. Flournoy, W.F. Rosenberger, and W.K. Wong, "Global versus local search in constrained optimization of computer models," In book: New developments and applications in experimental design (pp.11-25), Institute of Mathematical Statistics, Hayward, CA, 1998.
[44]. S. Bagheri, W. Konen, R. Allmendinger, J. Branke, and K. Deb, et al., "Constraint Handling in Efficient Global Optimization," In Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017.
[45]. H. Dong, B. Song, Z. Dong, and P. Wang, "SCGOSR: Surrogate-based constrained global optimization using space reduction," Applied Soft Computing, vol.65, pp.462–477, 2018.
[46]. Y. Wu, Q. Yi, H. Jie, B. Wang, and J. Zhao, "A RBF-based constrained global optimization algorithm for problems with computationally expensive objective and constraints," Structural and Multidisciplinary Optimization, vol.58, pp.1633–1655, 2018.
[47]. S. Bagheri, W. Konen, and T. Bäck, "Equality Constraint Handling for Surrogate-Assisted Constrained Optimization,"2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, July 2016.
[48]. R.G. Regis and S.M. Wild, "CONORBIT: constrained optimization by radial basis function interpolation in trust regions," Optimization Methods & Software, vol. 32, no.3, pp.552–580, 2017.
[49]. R.G. Regis, "Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions," Computers &Operations Research, vol.38, pp. 837–853, 2011.
[50]. I. Chung, D. Park, and D. Choi, "Surrogate-based global optimization using an adaptive switching infill sampling criterion for expensive black-box functions," Structural and Multidisciplinary Optimization, vol. 57, pp.1443–1459, 2018.
[51]. H. Dong, B. Song, Z. Dong, and P. Wang, "Multi-start Space Reduction

(MSSR) surrogate-based global optimization method," Structural and Multidisciplinary Optimization, vol.54, pp.907–926, 2016.
[52]. Z. Ugray, L. Lasdon, and J. Plummer et al., "Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization," Informs Journal on Computing, vol.19, no.3, pp. 328–340, 2007.
[53]. J. O. H. Sendin, J. R. Banga, and T. Csendes, "Extensions of a Multistart Clustering Algorithm for Constrained Global Optimization Problems," Industrial& Engineering Chemistry Research, vol.48, no. 6, pp. 3014–3023, 2009.
[54]. H. Dong, S. Sun, B. Song, and P. Wang, "Multi-surrogate-based global optimization using a score-based infill criterion," Structural and Multidisciplinary Optimization, vol.59, pp.485–506, 2019.
[55]. J. Gu, G.Y. Li, and Z. Dong, "Hybrid and adaptive meta-model-based global optimization," Engineering Optimization, vol. 4, no. 1, pp. 87–104, 2012.
[56]. T. Long, D. Wu, and X. Guo et al., "Efficient adaptive response surface method using intelligent space exploration strategy," Structural and Multidisciplinary Optimization, vol.51, no.6, pp.1335–1362, 2015.
[57]. C. E. Rasmussen and C. K. I. Williams, "Gaussian Processes for Machine Learning," MIT Press, Cambridge, USA, 2006.
[58]. S. Bagheri, W. Konen, and C. Foussette et al., "SACOBRA: Self-adjusting constrained black-box optimization with RBF," in Proc. 25. Workshop Computational Intelligence, F. Hoffmann and E. Hüllermeier, Eds. Universitätsverlag Karlsruhe, pp. 87–96, 2015.
[59]. S. Sakata, F. Ashida, and M. Zako, "An efficient algorithm for Kriging approximation and optimization with large-scale sampling data," Computer Methods Application Mechanics and Engineering, vol.193, pp.385–404, 2004.
[60]. B. Schölkopf and AJ. Smola, "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond," MIT Press, Cambridge, USA, 2002.
[61]. T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," IEEE Transactions on Evolutionary Computation, vol. 4, no.3, pp.284–294, 2000.
[62]. T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 35, no.2, pp. 233–243, 2005.
[63]. M. M. Efrén and C.D. Omar, "Empirical analysis of a modified Artificial Bee Colony for constrained numerical optimization," Applied Mathematics and Computation, vol. 218, pp.10943–10973, 2012.
[64]. P.A. Simionescu and D. Beale, "New Concepts in Graphic Visualization of Objective Functions," Information in Engineering Conference. Montreal, Canada. pp. 891–897. Retrieved 7 January 2017.
[65]. https://en.wikipedia.org/wiki/Test_functions_for_optimization#cite_note-13. Mishra's Bird function constrained.
[66]. S.N. Lophaven, H.B. Nielsen, and J. Sindergaard, "Aspects of the Matlab Toolbox DACE," Report IMM-REP-2002-13, Informatics and Mathematical Modelling, DTU. (2002), 44 pages. Available as http://www.imm.dtu.dk/»hbn/publ/TR0213.ps
[67]. Y. Li and Y. Wu, et al., "A Kriging-based constrained global optimization algorithm for expensive black-box functions with infeasible initial points," Journal of Global Optimization, vol. 67, no.(1-2), pp. 343-366, 2017.
[68]. T. Tao, G. Zhao and S. Ren, "An Efficient Kriging-Based Constrained Optimization Algorithm by Global and Local Sampling in Feasible Region," ASME. J. Mech. Des., vol. 142, no.5, pp. 051401, 2020.
[69]. Y. Li, Y. Wu, et al., "A Kriging-based bi-objective constrained optimization method for fuel economy of hydrogen fuel cell vehicle," International Journal of Hydrogen Energy. vol.44, no.56, pp.. 29658-29670 , 2019
[70]. C. Peng, H. Liu and E. D. Goodman, "A Cooperative Evolutionary Framework Based on an Improved Version of Directed Weight Vectors for Constrained Multiobjective Optimization With Deceptive Constraints," IEEE Transactions on Cybernetics, doi: 10.1109/TCYB.2020.2998038.

Yuanping Su received the bachelor's degree in automation in 2002, and the master's degree in control theory and control engineering in 2009 from the Jiangxi University of Science and Technology, Ganzhou, China, and Ph.D in control theory and control engineering in 2016 from Tongji University, Shanghai, China. He is currently working in Jiangxi University of Science and Technology as an associate professor, and from Aug. 1, 2018, he has been working at Michigan State University as a postdoctor. His current research interests include nonlinear system control theory, evolutionary optimization, and modeling and control of greenhouse environment.

Lihong Xu received the Ph.D degree in engineering from the Department of Automatic Control, Southeast University, Nanjing, China, in 1991.In 1994, he was appointed as a Professor at Southeast University. He transferred to Tongji University in August, 1997, and has been a Professor with Tongji University since then. His research fields include control theory, computational intelligence, and optimization theory. In 1998 he was funded by the Daylight Project of Shanghai. In 1999 he was funded by the University Backbone Young Tutors Project of Ministry of Education of China. He got a National Science and Technology Progress Award (second prize) of China in 2007. He is now doing joint research work as a Visiting Professor and Advisor of the greenhouse research team of BEACON, USA.

Prof. Xu is a member of ACM, a senior member of IEEE and the President of IEEE CIS's Shanghai Chapter. He was the Co-Chair of the 2009 GEC Summit in Shanghai. He is also a Standing Director of the Chinese Society of Agricultural Engineering and the President of Shanghai Society of Agricultural Engineering

Erik D. Goodman received the B.S. degree in mathematics and the M.S. degree in system science from Michigan State University (MSU), East Lansing, MI, USA, and the Ph.D. degree in computer and communication sciences from the University of Michigan, Ann Arbor, MI, USA, in 1972.He is Executive Director of the BEACON Center for the Study of Evolution in Action, an NSF Science and Technology Center, MSU. He has been a Professor with the Department of Electrical and Computer Engineering, MSU, since 1984.He is the Co-Founder in 1999 and the former Vice President for Technology of Red Cedar Technology, Inc., where he authored the SHERPA algorithm now offered by Siemens PLM, the owner of Red Cedar Technology. He is an Advisory Professor with Tongji University, Shanghai, China, and East China Normal University, Shanghai. Dr. Goodman is a Senior fellow of the International Society for Genetic and Evolutionary Computation. He was the Chair of the Executive Board of that society from 2001 to 2004. From 2005 to 2007, he was the Founding Chair of the ACM SIG for Genetic and Evolutionary Computation, SIGEVO.