International Journal of Computational Intelligence Systems
Vol. **13(1)**, 2020, *pp.* 1663–1678
DOI: https://doi.org/10.2991/ijcis.d.201014.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

A New Approach for Low-Dimensional Constrained Engineering Design Optimization Using Design and Analysis of Simulation Experiments

Amir Parnianifard[1,*], Ratchatin Chancharoen[2], Gridsada Phanomchoeng[2], Lunchakorn Wuttisittikulkij[1,*]

[1]*Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand*
[2]*Department of Mechanical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand*

ARTICLE INFO

Article History
Received 17 May 2020
Accepted 06 Oct 2020

Keywords
Constrained optimization
Surrogates
Kriging
Computationally expensive function
Global optimization

ABSTRACT

The number of function evaluations in many industrial applications of simulation-based optimization problems is strictly limited. Therefore, only little analytical information on objective and constraint functions is available. This paper presents an adaptive algorithm called the Surrogate-Based Constrained Global-Optimization (SCGO) method to solve black-box constrained simulation-based optimization problems involving computationally expensive objective function and inequality constraints. Firstly, Kriging surrogate is constructed over a new overall objective function (called loss function) to approximate the behavior of a true model. Then, an adaptive approach is provided to improve the optimal results sequentially while enforcing a feasible solution. The SCGO method is tested on several classical engineering design problems namely design of a tension/compression spring, design of a welded beam, design of a pressure vessel, and three-bar truss design. The results demonstrate that SCGO has advantages in solving the costly constrained problems and needs less costly function evaluations. Optimization results prove that the proposed algorithm is very competitive compared to the state-of-the-art metaheuristic algorithms.

1. INTRODUCTION

Optimization technologies have been extensively applied in engineering design problems to improve the design quality and shorten the design cycle. In engineering and mathematics, optimization problems based on mathematical characteristics consist of three basic elements including (i) an objective function that needs to be minimized or maximized, (ii) constraints that are relations between decision variables and the parameters, (iii) the set of controllable inputs (decision variables) which affects the value of the objective function. In an optimization problem, the types of mathematical relationships between the objective and constraints and the decision variables determine how difficult it is to solve, the solution methods or algorithms that can be used for optimization, and the assurance of the optimality of the solution. Optimization problems can be categorized into many classes depending on the linearity of the objective function, the modality of the objective function, the number of objective functions, the availability of constraints, the number of decision variables, and the linearity of the constraints (discrete or continuous). A general optimization problem is usually described as a combination of these classifications, see [1]. Generally, the term optimization means to find the best levels of design variables set (X) according to one or multi- objective(s) $f(X)$ while keeping design variables within their constraints $c(X)$. Such constraints can be designed by equalities or inequalities which cause the design space to be limited for searching the best solution. However, a general framework in the mathematical programming model can be depicted as

$$\begin{aligned} &Min\ or\ Max: \quad f(X) \\ &Subject\ to: \quad c_i(X) \le 0,\ \ i = 1, 2, \ldots, m \end{aligned} \tag{1}$$

where $f(X)$ shows the objective function and $c_i(X)$ illustrates the set of inequality constraints [2]. Since most engineering design problems are limited to constraints, it is crucial to have an appropriate strategy to handle the design constraints [3,4]. In general, optimization techniques that try to find an optimum and feasible solution by covering constraints' requirements are classified into model-based and surrogate-based (also known as metamodel based) approaches [5,6]. Metaheuristics optimization techniques have become very popular in the last two decades. Remarkably, some of them such as Genetic Algorithm (GA) [1,7], Ant Colony Optimization (ACO) [8], and Particle Swarm Optimization (PSO) [9] are the most well-known not only among computer scientists but also scientists from different fields [10]. In addition to a large number of theoretical-based studies, such optimization techniques have been applied in various fields of study. Here, the question is why metaheuristics have become remarkably common. The answer to this question can be summarized into four main reasons that are simplicity,

*Corresponding authors. Email: Lunchakorn.W@chula.ac.th; amir.p@chula.ac.th

flexibility, derivation-free mechanism, and local optima avoidance [10–12]. However, for "expensive optimizations" such as computer-aided design optimization or simulation-based optimization using computer experiments with high computational cost (i.e., a high number of function evaluations to solve the problem), traditional model-based optimization methods may perform poorly or may even fail to obtain a satisfactory solution within the allocated computational budget. To avoid this, researchers turn to computational intelligence methods such as agent-based algorithms [13], fuzzy logic [14,15], artificial neural networks [16], and surrogates techniques [17–20].

For the past decades, lots of practical engineers and scientists have not trusted expensive model-based optimization methods to obtain designs [21]. As opposed to the algebraic model-based models, surrogate-assisted simulation-optimization does not assume that an algebraic description of the true model is available. However, the model may be available as a black-box that only allows the approximation of the objective and constraints as input/output sets of data. Also, many large-scale and/or detailed true-models may be time-consuming to run [22–26]. To overcome such difficulties, researchers have applied surrogate-based learning methods (e.g., polynomial regression, Kriging, and radial basis function) [18,27,28]. Surrogate-based methods can "learn" the problem behaviors and approximate the function values. These approximation models can speed up the function evaluation and the estimation of the function value with acceptable accuracy. Besides, they can improve the optimization performance and provide a better final solution. Various types of real-world engineering optimization problems have been solved by applying surrogate-based methods such as dynamic and stochastic control system design, sub-communities in machine learning problems, discrete event systems (e.g., queues, operations, and networks), manufacturing, medicine and biology, engineering, computer science, electronics, transportation, and logistics, see [17–19,25,29,30]. Several studies have systematically illustrated the advantages of surrogate-based optimization algorithms [18,21,23,31].

1.1. Related Works

A complex engineering model often takes several or more hours to run a single simulation. Also, design problems in engineering applications as described by the simulation models, are always "black-box" [32]. The search in constraint black-box optimization is difficult since there is not usually a priori knowledge about the feasible region and the fitness landscape. This problem becomes more complex when only a limited number of function evaluations are allowed to be used in the search. The area of efficient constrained optimization is optimization under a severely limited budget of fewer than 1000 functions [33]. Due to the high-computational overhead and the unknown function expressions, common optimization algorithms including Whale Optimization Algorithm (WOA) [11], Grey Wolf Optimizer (GWO) [10], Gravitational Search Algorithm (GSA) [34], Salp Swarm Algorithm (SSA) [35], Co-evolutionary Particle Swarm Optimization (CPSO) [36], Grasshopper Optimization Algorithm (GOA) [37], and Ant Lion Optimizer (ALO) [12] are very expensive if they are directly implemented on these computationally expensive models. For instance, these algorithms require 4,000–18,000 function evaluations to solve the classical constrained engineering design problems namely design of a tension/compression spring, design of a welded beam, design of a pressure vessel, and three-bar truss design. The most common way to incorporate constraints into the mentioned metaheuristics is to employ penalty functions [38,39]. There are also several other approaches available for constraint handling, see review on constraint optimization presented by [38,40,41].

These real-world applications commonly belong to the class of black-box simulation-optimization problems. The state-of-the-art in handling complex and costly problems arising from real-world applications using surrogate models in optimization is reviewed in [3,17–20]. The Efficient Global Optimization (EGO) algorithm was first introduced by [42] that used surrogate models (mostly Kriging) through maximizing Expected Improvement (EI) criterion. This algorithm has been developed in different studies, see [43–45]. Also, the probability of improvement [46], the generalized EI [47], and the augmented EI [48] have been also proposed to adaptively determine the next sampling point in Kriging-based optimizations. A surrogate-based gradient-free optimization algorithm has been developed in [32] that can handle the optimization problems including nonlinear equality or inequality constraints, where the function evaluations are expensive. A surrogate-based PSO method to solve structural design optimization problems with expensive constraint functions has been presented in [49]. RBF-assisted evolutionary programming algorithm to solve high-dimensional problems with the black-box objective and inequality constraints has been developed in [32,33,50]. Simpson *et al.* summarized the use of each metamodel. RSM is well established, easy-to-use, and suitable for low dimensions. Neural Network metamodel is more suitable for highly nonlinear problems and is recommended for deterministic applications. Kriging metamodel is suitable for low dimension and it is flexible regarding employing different correlation functions [17,51]. Many Kriging-assisted efficient optimizers have been developed in the last years [52–54]. However, Kriging's performance is well confirmed in low-dimensional problems regarding the scope of the current study, see [23,28,55,56].

1.2. Problem Statement

Regarding the limitations found in existed common constrained optimization studies, this study consists of three main contributions as follows:

- Most optimization algorithms need to set their parameters concerning the specific optimization problem to show good performance. In other words, it has been necessary for such methods to carefully adjust the parameters of the algorithm to each problem. In real black-box optimization, all these adjustments would probably require knowledge of the problem or several executions of the optimization code [33]. There are different parameters that have been manually adjusted in [57–59] such as distance requirement cycle, constraint normalization, random start probability, and logarithmic transformation of the fitness function, see [33]. The first contribution of the current study is to present the algorithm which does not need manual adjustment of the mentioned parameters to the problem at hand.

- Generally, sequential sampling methods are used to iteratively update the surrogate and improve the solution found during the optimization process. These methods iteratively refit the surrogate to improve the best solution so far during the optimization process. However, the updating procedure increases computational time-consumption. Moreover, one of the main contributions of the proposed algorithm is that only one surrogate is fit on the initial training sample points, and there is no need to refine the surrogate sequentially.
- Various surrogate-based optimization algorithms are developed to solve problems with expensive simulations. However, deciding on the suitability of the surrogate model to be incorporated within a constrained optimization model remains an open challenge [21,26,60]. There is also a lack of studies in the literature that compare the performances of surrogate's application and metaheuristics in the same platforms of constrained optimization problems. The proposed method is developed to employ the small number of function evaluations as well as to keep the accuracy of the optimal results (lower objective function value) as compared to some state-of-the-art metaheuristic algorithms through the same four platforms of constrained engineering design problems.

This paper aims to develop an effective search algorithm based on surrogate applied in the optimization of expensive simulation models (i.e., with costly function evaluation). This paper limits its focus to low-dimensional (small-scale) constrained optimization models with less than five design variables with a single objective. Space-filling design methods such as Latin Hypercube Sampling (LHS) and Grid Design (GD) [61,62] are employed to construct both training and search sample points. A new overall loss function is introduced by integrating the normalized single objective and constraints of the optimization model. Then, one surrogate is constructed on this overall loss function. This surrogate can be effectively used to create cheaper outputs using larger sample points to speed up the search of the global optimum with a limited number of function evaluations and less computational time. Using sorted grid points, the relevant true model's output is computed and the best feasible point is obtained. Then, an adaptive improvement approach is derived to sequentially improve the obtained solution and reach an optimal point.

The rest of this paper is organized as follows: Section 3 describes the proposed approach developed in this study. Test problems for constrained engineering design problems in the mathematical structure and their experimental results are presented in Section 3. Section 4 provides more discussions for the obtained results. Finally, this study is concluded in Section 5.

2. PROPOSED ALGORITHM

In this section, the proposed approach namely Surrogate-Based Constrained Global-Optimization (SCGO) is described which solves the constrained optimization problem.

$$\begin{aligned} &Min\ \ f(x) \\ &x \in \omega \\ &Subject\ to:\ \ c_1(x) \le 0, c_2(x) \le 0, \ldots, c_m(x) \le 0 \end{aligned} \tag{2}$$

where $x \in \omega$ defines the design space. Practical problems in real systems are mostly constrained. There are two types of constraints including inequality and equality involved in defining the feasibility of solutions when designing of the optimization model. Equality constraints can be considered as particular cases where $c_i(x) \le 0$. For optimizing constrained problems, a constraint handling method should be integrated into the optimizer. There are several methods of constraints handling in the literature including penalty functions, special operators, repair algorithms, separation of objectives and constraints, and hybrid methods, see [3,38]. Here, to tackle hard computational-constrained problems, inspired by the Taguchi quality loss function [63,64], an overall loss function is proposed by integrating all constraints and objective functions.

2.1. Algorithmic Framework

Figure 1 sketched the flow diagram of the proposed SCGO approach. The main steps involved in the proposed algorithm proceed in the following steps. Also, the pseudocode of SCGO is presented in Algorithm 1.

Step 1: *Design training sample points*

In this paper, the SCGO algorithm is developed for the low-dimensional (i.e., less than four variables) and expensive problems. For such problems, it has been recommended to design 30 training points as a small sample set or 100 training points as a large sample set using space-filling approaches (e.g., LHS, GD), see [56,65]. The space-filling approach can treat all the regions of design space equally. LHS has been commonly defined for designing sample points based on the space-filling concept [66,67]. Undoubtedly, the simplest and the most popular sampling design methods utilize LHS [17,18].

Three common choices are available to ensure appropriate space-filling of the sample points in LHS design namely minimax,

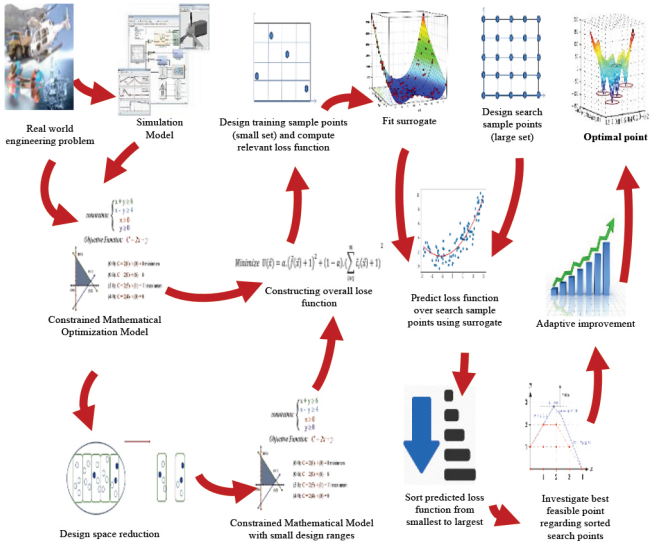

Figure 1 The procedure of proposed Surrogate-Based Constrained Global-Optimization (SCGO) approach.

Algorithm 1: Proposed SCGO algorithm in pseudocode.

Input: Objective function f, set of constraint function(s), $C = (c_1, c_2, \dots, c_m)$ where m is the number of constraints in a model, $\vec{x} \in \omega$ where ω is design space, the number of design variables n_x, the weight scale α where $0 \le \alpha \le 1$, see Eq. (3)

Output: The optimal solution $(\vec{x}_{opt}, f_{opt})$ found by the algorithm.

1: **begin**

2: Design initial training sample points using space-filling design (e.g. LHS) with dimension $(l \times n_x)$ where l is the number of training sample points (input combinations).

3: Run the simulation model and obtain f_s and $c_{is}\ \forall s = 1, 2, \dots, l$ and $\forall i = 1, 2, \dots, m$

4: Normalize objective function to $[-1, +1]$:

$$\tilde{f}_s(\vec{x}) \leftarrow \left\{2.\left(\frac{f_s(\vec{x}) - \min_s f(\vec{x})}{\max_s f(\vec{x}) - \min_s f(\vec{x})}\right) - 1\right\}$$

5: Normalize all constraint functions to $[-1, +1]$:

$$\tilde{c}_{is}(\vec{x}) \leftarrow \left\{2.\left(\frac{c_{is} - \min_s c_i(\vec{x})}{\max_s c_i(\vec{x}) - \min_s c_i(\vec{x})}\right) - 1\right\} \forall i = 1, 2, \dots, m$$

6: Compute overall loss function $U(\vec{x})$ for each input combination l:

$$U_s(\vec{x}) = \alpha.\left(\tilde{f}_s(\vec{x}) + 1\right)^2 + (1-\alpha).\left(\sum_{i=1}^{m} \tilde{c}_{is}(\vec{x}) + 1\right)^2$$

7: Put s as an input set, $U_s(\vec{x})$ as an output set, and construct Kriging surrogate over set of input/output set of data.

8: Design search sample points using grid sampling design with dimension $(L \times n_x)$ where $L \gg l$, and L is the number of search sample points.

9: Run Kriging and predict $\hat{U}_h$ for $h = 1, 2, \dots, L$

10: **Sort** $(\hat{U}_h, \vec{x}_h)$ ▶ sort regarding smallest to largest loss function.

11: Set $n = 0$.

12: **while** (obtain $n = k$) **do**

▶ k best feasible points in set of sorted $\vec{x}_h$

13: Run simulation model for first point in set of sorted $\vec{x}_h$ and check feasibility of point.

14: **if** (point is feasible) **then**

15: $n = n + 1$, and $\vec{x}_b^{best} \leftarrow \vec{x}_h$ where $b = 1, 2, \dots, n$

16: **else if**

17: Run simulation model for the next point in set of sorted $\vec{x}_h$ and check feasibility of point.

18: **end if**

19: **end while**

20: **for** (all $\vec{x}_b^{best}$) **do**

21: Run adaptive improvement

▶ see Algorithm 2.

22: $\vec{x}_{opt} \leftarrow$ select the point with minimum fitness function after adaptive improvement

23: **end for**

24: **return** $(\vec{x}_{opt}, f_{opt})$

25: **end algorithm**

maximin, and desired correlation matrix, see [28]. The GD design can fill the whole design space like LHS and has the strength to allocate points in each corner of the design space. The GD was adapted to balance discrete experimental factors in a continuous space [68]. In this study, two Matlab® functions namely "lhsamp" and "gridsamp" in the DACE toolbox are used to design the required sample points.

Step 2: *Run the true model and collect the model's outputs*

Regarding the designed training sample points in the previous step, the true model (constrained mathematical model) is run and relevant outputs are computed for the objective function, and each constraint function separately.

Step 3: *Compute overall loss function*

To define an overall function to be used in search of an optimal feasible point, an overall loss function is proposed by the following equation:

$$\begin{aligned} Minimize\ U(\vec{x}) = \alpha.\left(\tilde{f}(\vec{x}) + 1\right)^2 \\ + (1-\alpha).\left(\sum_{i=1}^{m} \tilde{c}_i(\vec{x}) + 1\right)^2 \end{aligned} \tag{3}$$

This loss function is computed for each training sample point (input combination) using collected outputs from Step 2 in the objective function and constraints. In this loss function, the weight factor $0 \le \alpha \le 1$ shows a preference between objective function and constraints set in the search procedure. This weight is defined by a decision-maker. The parameter α in Eq. (3) enforces the search procedure to investigate the point with a lower objective function or the point with a higher chance of feasibility. In Eq. (3), $\tilde{f}(x)$ depicts the normalized objective function value in the range $[-1, 1]$ and $\tilde{c}_i(x)$ illustrates the normalized constraint value for ith constraint function in the range $[-1, 1]$. All the values for objective and constraint functions that obtained over-designed sample points (previous steps) are normalized in $[-1, 1]$ by the following equation:

$$y' = 2.\left(\frac{y - y_{min}}{y_{max} - y_{min}}\right) - 1 \tag{4}$$

where y_{max} and y_{min} are the maximum and minimum objective values of the designed training sample points separately. The rescaling in $[-1, 1]$ needs to be done for an objective function $\tilde{f}(x)$ and for each constraint $\tilde{c}(x)$.

Step 4: *Fit a Kriging surrogate*

In this step, a Kriging surrogate is fitted on computed input/output data in previous steps (designed training sample points by space-filling design and overall loss function). In a Kriging model, a combination of a polynomial model and realization of a stationary point is assumed by the form of

$$y = f(X) + Z(X) + \varepsilon \tag{5}$$

where $f(X) = \sum_{j=0}^{k} \hat{\beta}_j f_j(X)$, the polynomial terms of $f_j(X)$ are typically the first or second-order response surface approaches and coefficient $\hat{\beta}_j$ is regression parameters $(j = 0, 1, \dots, k)$. The term ε describes approximation error and the term $Z(X)$ represents the realization of a stochastic process which is normally distributed by Gaussian random process with zero mean and variance σ^2, and nonzero covariance. The correlation function of $Z(X)$ is defined by

$$Cov\left[Z(x_i), Z(x_p)\right] = \sigma^2 R(x_i, x_p) \tag{6}$$

where σ^2 is process variance and $R(x_i, x_p)$ is the correlation function that can be chosen from different functions proposed in the literature. Due to tuning the correlation function with sample data, Kriging is extremely flexible to capture nonlinear treatment of the model. In the literature, some studies have been found which sufficiently describe Kriging methodology, see [69,70]. Here, "DACE," MATLAB® toolbox [71] is used to construct Kriging surrogate. In our study, the first-order polynomial function and Gaussian correlation function are assumed as two summation terms in Eq. (5).

Step 5: *Design search sample points*

One of the simplest approaches to the optimization method to search for optimal solutions by employing surrogate instead of an expensive original model is the "grid-search" method. In this approach, the whole design space (exploration) is investigated through the evaluation model components consisting of objectives and constraints sets in the equally spaced grid points [72]. Also, the grid points can be replaced with random points, which are spread in the whole design space (Monte Carlo random search) or with some other space-filling methods. In this searching method, there is a chance to find the global optimum solution because searching is not bounded to the local valley [23]. Here, a space-filling design method is used to produce two groups of sample points including training samples and search sample points. Training sample points are used for constructing surrogates (see Step 1). Search sample points are used to investigate the optimal feasible points. In the search procedure, the surrogate is run instead of the original constrained optimization model. Therefore, a greater set of sample points does not imply a computationally expensive task. Moreover, the greater set of sample points can be designed by a space-filling design method to derive the search procedures.

Step 6: *Approximate sorted loss function values*

In this step, the fitted Kriging surrogate (Step 4) is used to approximate the loss function values for each designed search sample point (Step 5). Then these values are sorted from smallest to largest.

Step 7: *Search optimal feasible points*

Regarding the sorted search points that are arranged from the smallest to the largest predicted loss values (see Step 6), the original true model (here mathematical constrained optimization model) is run and the relevant values of the constraint functions are obtained. It starts from the smallest predicted loss value. All the infeasible points are removed from the list of candidacy (i.e., death penalty method for constraint handling, see [38]). For the first obtained feasible point, the relevant objective value is computed. Here, the three first best feasible points are considered to derive adaptive procedure on them as elucidated in the next step.

Step 8: *Derive adaptive improvement procedure on the best feasible points*

Sometimes, due to the low correlation between the predicted optimum point (which is estimated by surrogate) and the other points, the surrogate does not show enough accuracy in the optimal point as compared to the original model. For instance, Figure 2 shows a shifting predicted optimal point comparing with a true optimum point in an original model for a problem with two input variables. It means that the surrogate is not able to detect the location of the true optimum point when no points are sampled around the true optimum point. The appropriate updating procedure needs to be done to improve the optimal result sequentially, particularly around the predicted optimum points. This is obtained by adding new neighbor points around the current best point. However, to find the neighbor points around the current best point, different strategies of sampling design can be employed (e.g., full factorial design, fractional factorial design, etc.). For models with a small number of design variables (low-dimensional constrained optimization problem), two-level full factorial and fractional factorial designs have been recommended [73,74] to define t candidates of infill points around the current best point. If $j = (1, 2, \ldots, n)$ shows the number of design variables in the model, based on the two-level full factorial method, the number of candidates in the vicinity of the best point is $t = 2^j$. If x_j^b denotes the value of j^{th} design variable in the current best sample point, the upper level (u_j^b) and lower level (l_j^b) for each design variable based on two-level full factorial design around the center (x_j^b) can be computed by

$$u_j^b = x_j^b (1 + \theta) \qquad (7)$$
$$l_j^b = x_j^b (1 - \theta)$$

where parameter θ, $(0 \leq \theta \leq 1)$ controls the distance between candidates and the center's location regarding two-level full factorial design (i.e., the current best point located in the center). Besides, by changing parameter θ, the algorithm can be controlled to design candidate points inside the bound of design space (predefined bounds for design variables in a problem). Figure 3 illustrates the two-level full factorial design for a model with two design variables when four candidate points are designed around the current best point. Algorithm 2 presents the pseudocode of an adaptive improvement procedure.

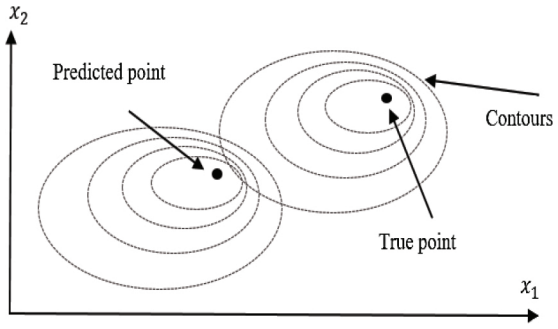

Figure 2 | True optimum versus predicted optimum with surrogate for design space with two variables.

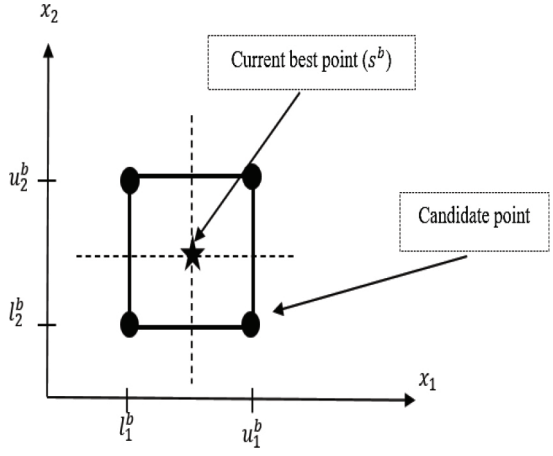

Figure 3 | Four candidate points around current best point based on the two-level full factorial design for problem with two design variables.

Algorithm 2: Adaptive improvement in pseudocode.

Input: The best feasible point $\vec{x}_b^{best}$ searched by Algorithm 1, the number of design variables n_x, $\vec{x} \in \omega$ where ω is design space, the weight scale θ where $0 \leq \theta \leq 1$, objective function f, set of constraint function(s), $C = (c_1, c_2, \ldots, c_m)$ where m is the number of constraints in a model, the number of sequential runs for adaptive improvement N_s
Output: The improved best feasible point $(\vec{x}_{imp}, f_{imp})$ found by the algorithm.
1: **begin**
2: **for** $n = 1 : N_s$ ▶ for $n = 1, \vec{x}_n \leftarrow \vec{x}_b^{best}$
3: **for** $t = 1 : 2^{n_x}$ ▶ design $t = 2^{n_x}$ sample points around $\vec{x}_n$ using a two-level full factorial design.
4: **while** $(\vec{x}_t \in \omega)$ **do**
5: Define/redefine the weight scale θ where $0 \leq \theta \leq 1$
▶ manually or randomly
6: $s_j \leftarrow x_j^b (1 \mp \theta) \forall j = 1, 2, \ldots, n_x$
7: $\vec{x}_t = [s_1, s_2, \ldots, s_j]$
8: **end while**
9: Run simulation model over $\vec{x}_t$ and compute f_t and $c_{it} \forall i = 1, 2, \ldots, m$.
10: **if** $(c_{it} \geq 0 \forall i = 1, 2, \ldots, m)$ **then** ▶ $\vec{x}_t$ is feasible
11: $O_t = f_t$
12: **else if**
13: $O_t =' NF'$
14: **end if**
15: $O_T(t, :) = O_t$ and $X_T(t, :) = \vec{x}_t$
16: **end for**
17: **if** (all members in O_T equal to $'NF'$) **then**
18: **break**
▶ return to line 2 and redefine θ
19: **end if**
20: $\vec{x}_n \leftarrow replace\, \vec{x}_n$ by the sample point with minimum f_t in $[O_T]$
21: $\vec{x}_N(n, :) = \vec{x}_b$ and $f_N(n, :) \leftarrow$ minimum f_t in $[O_T]$
22: end for
23: $\vec{x}_{imp} \leftarrow$ select the point with minimum fitness function in $[f_N]$
24: **Return** $(\vec{x}_{imp}, f_{imp})$
25: **end algorithm**

2.2. Space Reduction Procedure (If Needed)

For a surrogate model to be useful in an optimization context, the surrogate model must be accurate at the sequence of iterates generated by the search algorithm as it converges towards the true optimum. In the cases with wide design space, first, the size of the design space needs to be reduced to a smaller region of interest to allow for a more accurate surrogate model to be generated. Different algorithms have been developed for design space reduction such as variable-fidelity framework with design space reduction [75], a rough set method [76], and domain optimization algorithm [77]. Moreover, in such problems with wide design space, to gain more accuracy in the constructed surrogate, a design space in a smaller region needs to be reduced first. This pre-optimization procedure is performed before the main optimization procedure regarding Algorithm 1. The proposed design space reduction algorithm proceeds as follows:

Step 1: Design the experiments (using space-filling design method) for the original range of design space.

Step 2: Obtain the true model's output for the objective function and each constraint for designed sample points in the previous step.

Step 3: Fit a surrogate for input/output data for objective and one surrogate for each constraint separately (i.e. polynomial regression can be used as a cheaper surrogate than Kriging).

Step 4: Derive optimization procedure and obtain the best feasible point using common optimization methods (e.g. genetic algorithm or particle swarm optimization).

Step 5: Improve the result using an adaptive improvement procedure for the obtained best feasible point from the previous step and gain a rough optimum point.

Step 6: Define the smaller region of design space (compute the upper and the lower bounds) by w% around the obtained rough optimum point (where $0 \leq w \leq 1$ and is defined by decision-maker).

3. TEST OF THE SCGO METHOD IN ENGINEERING APPLICATION

In this section, four real-world constrained engineering design problems namely tension/compression spring, pressure vessel designs, welded beam, and three-bar truss design are employed to investigate the performance of the proposed method. Note that all of the four problems are performed in Matlab® environment. In the following, these test problems and given results using the proposed SCGO method are explained in detail.

3.1. Tension-Compression Spring Design

A tension-compression spring design problem is described in [78]. In this problem, the objective is to minimize the weight of a spring $(f(\vec{x}))$, see Figure 4. This problem includes three design variables namely the mean coil diameter ($D = [x_2]$), the wire diameter ($d = [x_1]$), and the number of active coils ($N = [x_3]$). The problem can be stated as

$$Consider\, \vec{x} = [x_1, x_2, x_3] = [d, D, N] \tag{8}$$

$$Minimize\, f(\vec{x}) = (x_3 + 2) x_2 x_1^2$$

$$Subject\ to:$$

$$c_1(\vec{x}) = 1 - \frac{x_2^3}{71785 x_1^4} \leq 0,$$

$$c_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566 (x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$$

$$c_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0,$$

$$c_4(\vec{x}) = \frac{x_2 + x_1}{1.5} - 1 \leq 0,$$

$$Variable\ range: 0.05 \leq x_1 \leq 2,$$

$$0.25 \leq x_2 \leq 1.30, 2 \leq x_3 \leq 15$$

This constrained optimization problem has been solved by the SCGO proposed method in this paper. In the steps of the problem-solving procedure, first, the training set is designed with the size of 30 sample points using the space-filling design (e.g., LHS, see [65]).

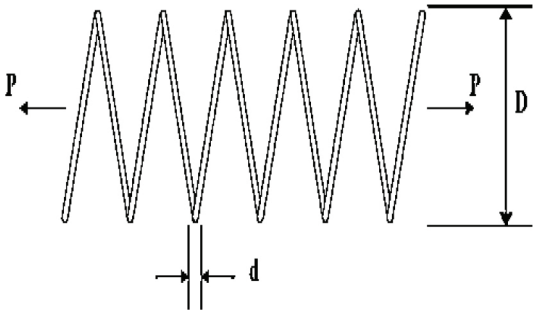

Figure 4 Schematic overview of tension/compression spring.

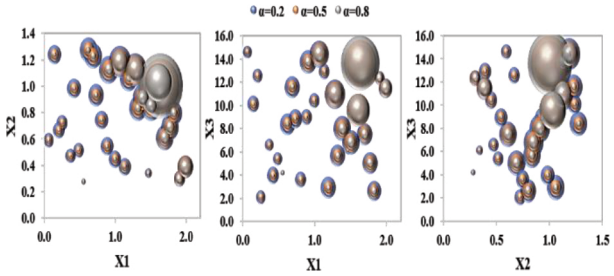

Figure 5 Magnitudes of loss functions over training sample points for two design variables in tension-compression spring design problem.

Then, for each sample point and regarding Equation (3), the overall loss function is computed for three different values of $\alpha = 0.2$, $\alpha = 0.5$, and $\alpha = 0.8$. Figure 5 shows the computed values of the overall loss function for the design space with two variables. Next, the overall loss function values are applied as an output set to construct the Kriging surrogate over the input set (designed training sample points). Figure 6 plots the fitted Kriging for two variables. To produce search sample points, the GD sampling method is used to design large size of sample points (e.g., for this problem 512,000 points are produced). The fitted Kriging surrogate is used to approximate the loss function relevant to each search sample point. Regarding the predicted loss function, sample points are sorted from smallest to largest. However, for the sorted sample points, the main mathematical optimization model is run and the relevant objective and constraints values are obtained. This search procedure is continued to obtain the first feasible point. Figure 7 (a) shows the feasible points regarding the main function evaluation in the sorted grid points (search sample points). Here, the first three feasible points are selected. Next, the adaptive improvement is performed for these three first feasible points, see Figure 7(b). The improvement procedure is derived for 30 sequential runs. In each run, 8 candidate points around the best point that have been obtained so far are designed (i.e., two-factorial designs for three variables are investigated). So, in this problem, a total of 240 function evaluations are done for adaptive improvement. This problem has also been solved by different methods including WOA [11], GWO [10], GSA [34], SSA [35], CPSO [36], GOA [37], and ALO [12]. The results of these studies are compared to those produced by the proposed approach are shown in Table 1. It can be seen that the best feasible solution found by SCGO is better than the best solutions found by the other techniques. In this problem, the SCGO can also converge to the same result in optimal weight 0.011172 for all three $\alpha = 0.2, 0.5$, and $\alpha = 0.8$. In the case of using the SCGO algorithm, the least number of function evaluations has been obtained by $\alpha = 0.8$ followed by $\alpha = 0.5$ and $\alpha = 0.2$, respectively. As per the maximum number of function evaluations reported in Table 1, the SCGO algorithm needs only 571 function evaluation to find a design with the optimal weight 0.011172 which is much less than other algorithms with more than 4,000 function evaluations.

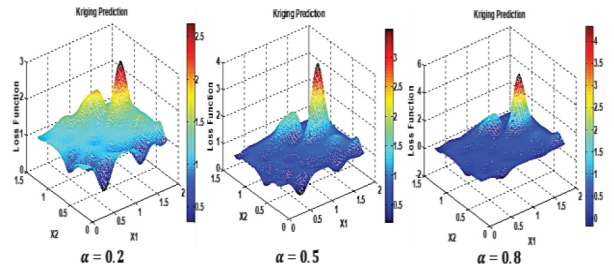

Figure 6 Surface plots of Kriging surrogate fitted over training points and loss functions for two design variables in tension-compression spring design problem.

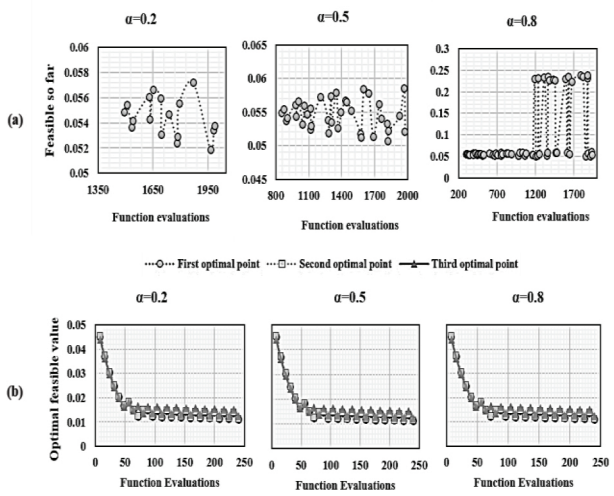

Figure 7 (a) Feasible points regarding sorted grid points (search sample points) and (b) adaptive improvement for three optimal points obtained from sorted grid points in tension-compression spring design problem.

3.2. Welded Beam Design

A welded beam is designed for minimum fabrication cost subject to constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c), end deflection of the beam (δ), and side constraints [79]. There are four optimization design variables as displayed in Figure 8 including the thickness of the weld (h), the length of the clamped bar (l), the height of the bar (t), and the thickness of the bar (b). This constrained optimization problem is mathematically formulated as follows:

Table 1 | Comparison of SCGO optimization results with literature for the tension-compression spring design problem.

Algorithm	Optimum Variables			Optimal Weight	Function Evaluations
	X1	X2	X3		
SCGO $\alpha = 0.2$	0.05074	0.36608	9.85518	0.011172	1763
SCGO $\alpha = 0.5$	0.05074	0.36608	9.85518	0.011172	1125
SCGO $\alpha = 0.8$	0.05074	0.36608	9.85518	0.011172	571
GWO	0.05169	0.35674	11.28885	0.012666	—
WOA	0.05121	0.34522	12.00403	0.012676	4410
GSA	0.05028	0.32368	13.52541	0.012702	4980
SSA	0.05121	0.34522	12.00403	0.012676	—
CPSO	0.05173	0.35764	11.24454	0.012675	5460

SCGO, Surrogate-Based Constrained Global-Optimization; WOA, Whale Optimization Algorithm; GWO, Grey Wolf Optimizer; GSA, Gravitational Search Algorithm; SSA, Salp Swarm Algorithm; CPSO, Co-evolutionary Particle Swarm Optimization.

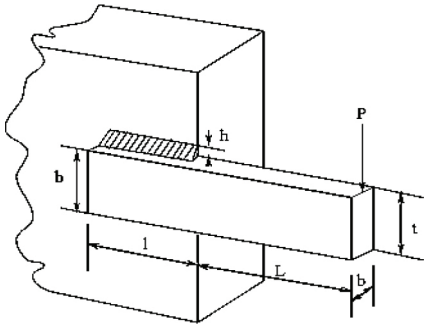

Figure 8 | Schematic overview of welded beam design problem.

$$Consider\ \vec{x} = [x_1, x_2, x_3, x_4] = [h, l, t, b] \tag{9}$$

$$Minimize\ f(\vec{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$$

$$Subject\ to:$$

$$c_1(\vec{x}) = \tau t(\vec{x}) - \tau t_{\max} = 0,$$

$$c_2(\vec{x}) = \sigma s(\vec{x}) - \sigma s_{\max} = 0,$$

$$c_3(\vec{x}) = x_1 - x_4 = 0,$$

$$c_4(\vec{x}) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 0.5 = 0,$$

$$c_5(\vec{x}) = 0.125 - x_1 = 0,$$

$$c_6(\vec{x}) = \delta(\vec{x}) - \delta_{\max} = 0,$$

$$c_7(\vec{x}) = P - P_c(\vec{x}) = 0,$$

$$Variable\ range: 0.1 = x_1 = 2, 0.1 = x_2 = 10,$$

$$0.1 = x_3 = 10, 0.1 = x_4 = 2$$

$$Where$$

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (t'')^2},$$

$$\tau' = \frac{P}{\sqrt{2}x_1 x_2},\ \tau'' = \frac{MR}{J},\ M = \left(L + \frac{x_2}{2}\right),$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

$$J = 2\left\{\sqrt{2}x_1 x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4 x_3^2}, \delta(\vec{x}) = \frac{4PL^3}{E x_3^3 x_4},$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000\,lb,\ L = 14\,in,$$

$$E = 30 \times 10^6\,psi,\ G = 12 \times 10^6\,psi$$

$$\tau_{\max} = 13,600\,psi,\ \sigma_{\max} = 30,000\,psi,$$

$$\delta_{\max} = 0.25\,in$$

To solve this constrained optimization problem with the proposed SCGO algorithm, the training sample points is designed with a size of 100 points. Figure 9 illustrates the magnitudes of the overall loss function in the set of training points. The optimization procedure is the same as for the previous problem, the Kriging surrogate is fitted over the input/output dataset for the design space with two variables, see Figure 10. The feasible points associated with the sorted grid points (i.e., sorted loss functions) are shown in Figure 11(a). Here, 480 function analyses are done for the adaptive improvement in 30 sequential runs. In each relevant sequential run, the candidate points are designed using a two-factorial design for four variables. Figure 11(b) illustrates this procedure for three feasible points obtained from the sorted grid points. This problem has also been solved by different methods including WOA [11], GWO [10], GSA [34], SSA [35], and CPSO [36]. The results are compared to those produced by the SCGO approach and are depicted in Table 2. This table shows that the optimal results obtained by SCGO outperform GSA and CPSO and provide very competitive results with GWO, WOA, and SSA. The result also shows the high performance of the SCGO algorithm in approximating the global optimum for this problem obtained in $\alpha = 0.8$ with cost function 1.757868. The results for both $\alpha = 0.2$ and $\alpha = 0.5$ converge to

cost function 1.859345. The maximum number of function evaluations in Table 2 also shows that the SCGO algorithm determines the global optimum for this problem with much less number of function evaluations than the other algorithms.

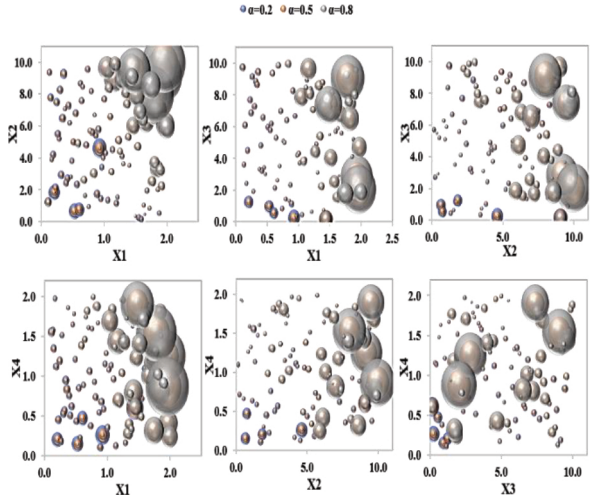

Figure 9 Magnitudes of loss functions over training sample points for two design variables in welded beam design problem.

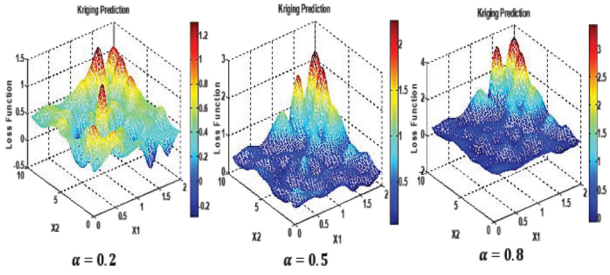

Figure 10 Surface plots of Kriging surrogate fitted over training points and loss functions for two design variables in welded beam design problem.

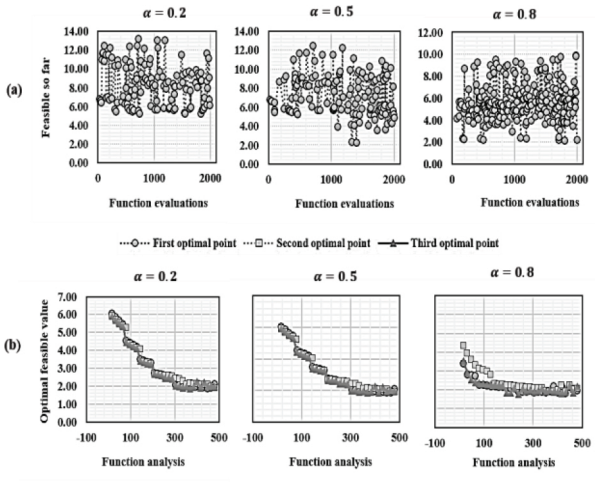

Figure 11 (a) Feasible points regarding sorted grid points (search sample points) and (b) adaptive improvement for three optimal points obtained from sorted grid points in welded beam design problem.

3.3. Pressure Vessel Design

The goal of this problem is to minimize the total cost $\left(f\left(\vec{x}\right)\right)$ including the cost of material, forming, and welding of a cylindrical vessel [80]. Both ends of the vessel are capped while the head has a hemispherical shape, see Figure 12. There are four design variables including the thickness of the shell (T_s), the thickness of the head (T_h), the inner radius (R), and the length of the cylindrical section without considering the vessel (L). The problem includes four constraints and can be stated as follows:

$$\begin{aligned}
&Consider\ \vec{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L] \qquad (10)\\
&Minimize\ f\left(\vec{x}\right) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2\\
&\qquad\qquad + 3.1661x_1^2x_4 + 19.84x_1^2x_3\\
&Subject\ to:\\
&c_1\left(\vec{x}\right) = -x_1 + 0.0193x_3 \le 0,\\
&c_2\left(\vec{x}\right) = -x_2 + 0.00954x_3 \le 0,\\
&c_3\left(\vec{x}\right) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \le 0,\\
&c_4\left(\vec{x}\right) = x_4 - 240 \le 0,\\
&Variable\ range:\ 0 \le x_1 \le 99, 0 \le x_2 \le 99,\\
&10 \le x_3 \le 200, 10 \le x_4 \le 200
\end{aligned}$$

In this problem, first, a space reduction procedure is performed to reduce the wide design space of the original model. For this purpose, the first 100 sample points are designed and for each sample, the relevant loss function is computed. For the obtained input/output data, a cheaper surrogate-like polynomial regression is fitted. This surrogate model is used to estimate the best feasible points that have been obtained so far. Then, an adaptive improvement (160 iterations) is derived for this point to improve the optimal result. The obtained values for each decision variable in this optimal point is applied to compute the upper and lower bounds for reduced design space by ±20% in these values. Notably, the Design-Expert® software is employed here for fitting polynomial regression. Also, its relevant optimization toolbox is used for performing the initial optimization toolbox in the procedure of design space reduction. Next, for the reduced design space of variables, the SCGO procedure is performed. The magnitudes of the overall loss function for a set of training points are shown in Figure 13. Figure 14 plots the 3D surface of the Kriging surrogate fitted on the training sample points (input) and the loss functions (output) set of data. The feasible points obtained over the sorted grid points are shown in Figure 15(a). Figure 15(b) illustrates the adaptive improvement for the three first best feasible points obtained by the sorted grid points. This problem has also been solved by different methods including WOA [11], GWO [10], GSA [34], and CPSO [36]. Table 3 compares the result produced by the proposed method in this paper with the results in the existing studies. The optimal solution found by SCGO is better than the solution found by GSA and provides competitive results with GWO, WOA, and CPSO algorithms. However, the best result by SCGO is obtained in $\alpha = 0.5$ with optimal cost 7157.687 and is followed by $\alpha = 0.8$ and $\alpha = 0.2$ with cost functions 7213.137 and 7684.383, respectively. In addition, it is observed the SCGO algorithm can find a design with the optimal cost with significantly lower number of function evaluations as compared to other GWO, WOA, GSA, and CPSO algorithms.

Table 2 Comparison of SCGO optimization results with literature for the welded beam design problem.

Algorithm	Optimum Variables			Optimal Cost	Function Evaluations
	X1	X3	X4		
SCGO $\alpha = 0.2$	0.18116	9.90000	0.20246	1.859345	628
SCGO $\alpha = 0.5$	0.18116	9.90000	0.20246	1.859345	635
SCGO $\alpha = 0.8$	0.20354	9.00000	0.20999	1.757868	695
GWO	0.20676	9.03681	0.20578	1.726740	—
WOA	0.20540	9.03743	0.20628	1.730499	9900
GSA	0.18213	10.00000	0.20238	1.879952	10750
SSA	0.20570	9.03660	0.20570	1.724910	—
CPSO	0.18290	9.36660	0.20590	1.824551	13770

SCGO, Surrogate-Based Constrained Global-Optimization; WOA, Whale Optimization Algorithm; GWO, Grey Wolf Optimizer; GSA, Gravitational Search Algorithm; SSA, Salp Swarm Algorithm; CPSO, Co-evolutionary Particle Swarm Optimization.

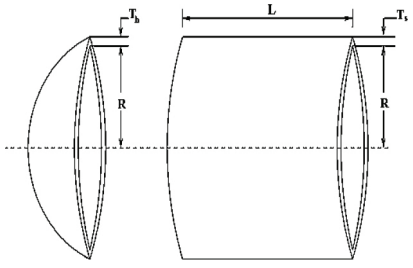

Figure 12 Schematic overview of pressure vessel design problem.

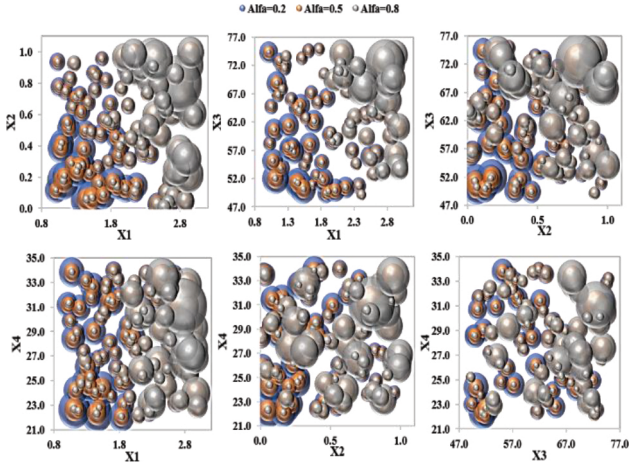

Figure 13 Magnitudes of loss functions over training sample points for two design variables in pressure vessel design problem.

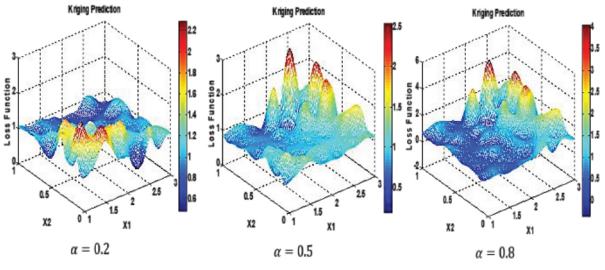

Figure 14 Surface plots of Kriging surrogate fitted over training points and loss functions for two design variables in pressure vessel design problem.

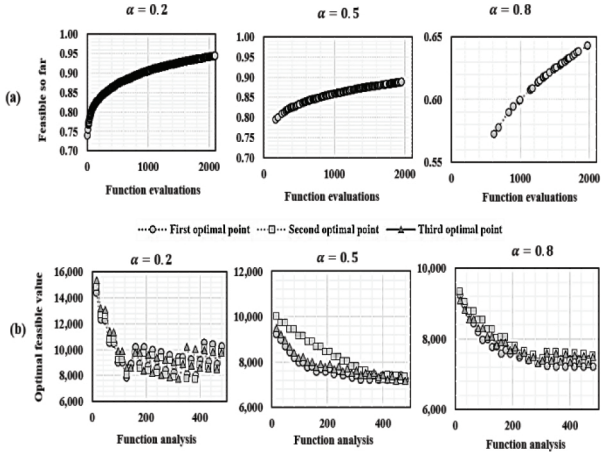

Figure 15 (a) Feasible points regarding sorted grid points (search sample points) and (b) adaptive improvement for three optimal points obtained from sorted grid points in pressure vessel design problem.

3.4. Three-Bar Truss Design Problem

This structural optimization problem exists in the field of civil engineering. This problem is widely utilized as case studies in the literature because of its highly constrained search space [35,37]. As shown in Figure 16, two parameters need to be optimally defined to achieve the least weight subject to stress, deflection, and buckling constraints. This problem is mathematically formulated as follows:

$$\begin{aligned}
&Consider\, \vec{x} = [x_1, x_2] = [A_1, A_2] \\
&Minimize f(\vec{x}) = \left(2\sqrt{2}x_1 + x_2\right) * l \\
&Subject\ to\ : \\
&c_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \le 0, \\
&c_2(\vec{x}) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \le 0, \\
&c_3(\vec{x}) = \frac{1}{\sqrt{2x_2 + x_1}} P - \sigma \le 0, \\
&Variable\ range\ : 0 \le x_1 \le 1, 0 \le x_2 \le 1, \\
&Where\ l = 100\,cm,\ p = 2\,KN/cm^2, \\
&\sigma = 2\,KN/cm^2
\end{aligned} \tag{11}$$

Table 3 | Comparison of SCGO optimization results with literature for the pressure vessel design problem.

Algorithm		Optimum Variables			Optimal Cost	Function Evaluations
		X1	X3	X4		
SCGO	$\alpha = 0.2$	1.26312	64.50464	13.28434	7684.383	849
	$\alpha = 0.5$	1.21502	62.09559	24.28143	7157.687	1097
	$\alpha = 0.8$	1.21502	62.09559	24.28143	7213.137	1452
GWO		0.81250	42.08918	176.75873	6051.564	—
WOA		0.81250	42.09827	176.63900	6059.741	6300
GSA		1.12500	55.98866	84.45420	8538.836	7110
CPSO		0.81250	42.09127	176.74650	6061.078	14790

SCGO, Surrogate-Based Constrained Global-Optimization; WOA, Whale Optimization Algorithm; GWO, Grey Wolf Optimizer; GSA, Gravitational Search Algorithm; SSA, Salp Swarm Algorithm; CPSO, Co-evolutionary Particle Swarm Optimization.

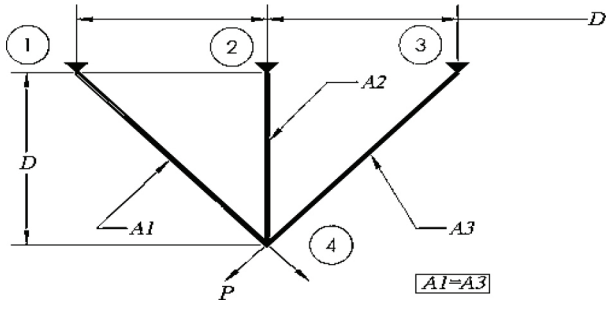

Figure 16 | Schematic overview of three-bar truss design problem.

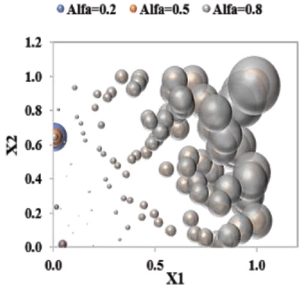

Figure 17 | Magnitudes of loss functions over training sample points for two design variables in three-bar truss design problem.

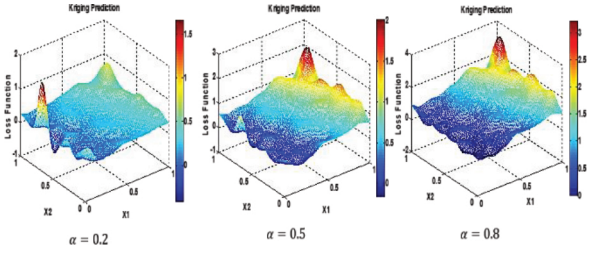

Figure 18 | Surface plots of Kriging surrogate fitted over training points and loss functions for two design variables in three-bar truss design problem.

This problem with two dimensions was solved by the SGCO algorithm. The magnitudes of the overall loss function and 3D surface plots constructed by Kriging are shown in Figures 17 and 18, respectively. Figure 19(a) and 19(b) illustrate the feasible points obtained by the sorted grid points and the adaptive improvement for the three first best feasible points obtained by the sorted grid points. The results of the proposed algorithm are compared to some common evolutionary algorithms to solve the three-bar truss problem (with the same platform) such as hybridizing Particle Swarm Optimization with Differential Evolution (PSO-DE) [81], GOA [37], and ALO [12] are shown in Table 4. The results indicate that very competitive results are provided by SCGO in obtaining optimal results in the three-bar truss design problem. The optimal cost functions by SCGO are obtained in $\alpha = 0.2$, $\alpha = 0.8$, and $\alpha = 0.5$ by optimal equals 264.33, 265.13, and 267.42, respectively. The smallest number of the function evaluations with a great difference belonging to the SCGO algorithm in $\alpha = 0.2$ with 222 function evaluations are compared to other solvers with 13,000, 14,000, and 17,600 function evaluations for GOA, ALO, and PSO-DE, respectively. Therefore, it can be understood that SCGO requires significantly less computational cost (very small number of function evaluations) to solve three-bar truss design problem with competitive accuracy (low objective function value) comparing with other state-of-the-art algorithms.

4. DISCUSSION

4.1. Performance Measure

In many studies on optimization, the strength of an optimization technique is measured by comparing the final solution achieved by different algorithms [33,82]. This approach only provides information about the quality of the results and neglects the speed of convergence which is a very important measure for expensive optimization problems. Comparing the convergence curve (number of function evaluations) is also one of the common benchmarking approaches [59]. However, a convergence curve provides good information about the final quality of the optimization result in terms of computational cost and it can be used to compare the performance of several algorithms only in one problem. Moré and Wild [83] have suggested performance measure for any pair (p, s) of problem p and solver s, to analyze the performance of any optimization algorithm as follows:

$$r_{p,s} = \frac{t_{p,s}}{\min\left\{t_{p,s'}\right\}}, \quad s, s' \in S \text{ and } p \in P \tag{12}$$

where P is a set of problems, S is a set of solvers, and $t_{p,s}$ is a number of function evaluations that solver $s \in S$ use to solve particular

problem $p \in P$. In Eq. (12), larger values of $t_{p,s}$ indicate a worse performance than other solvers. The convention $r_{p,s} = \infty$ is used when solver s fails to satisfy the convergence test on problem p.

However, Eq. (12) considers the required budget to solve the expensive optimization problem. In this study, inspired by [83], a new performance measure is used to consider two terms of the algorithm's performance including the level of accuracy and computational cost as follows:

$$R_{p,s} = \left\{ \beta \frac{t_{p,s}}{\min\left\{t_{p,s'}\right\}} + (1-\beta)\frac{l_{p,s}}{\min\left\{l_{p,s''}\right\}} \right\} \quad s, s', s'' \in S \text{ and } p \in P \tag{13}$$

where $l_{p,s}$ indicates the level of accuracy (i.e., lower objective function) for solver s in an expensive problem p, and γ ($0 \leq \beta \leq 1$) is the weight scale. Note that the best solver for a particular problem p attains the lower bound $R_{p,s} = 1$. In $\beta = 1$, the Eq. (13) provides the same performance measurement with suggested $r_{p,s}$ by [83] in Eq. (12). For $\beta = 0$, only the accuracy of solver in obtaining lower objective function is considered as a performance measure for comparing all the optimization algorithms and computational cost (number of function evaluation) is not considered. Finally, the performances of SCGO and other solvers for each problem have been measured by Eq. (13) when β varies in $[0,1]$. The results are shown in Figure 20. This figure reports a better performance $R_{p,s}$ of SCGO algorithm among all the four constrained engineering problems as compared to the other common optimizers in the literature, when the budget of optimization for expensive problems is limited to a small number of function evaluations. As can be seen from Figure 20, for β close to zero, all solvers have almost the same performance. By increasing β, the better performance of SCGO is highlighted more as compared to the other solvers, when they significantly lose their performances due to the required large number of function evaluations to solve the problem.

In addition to the desired performance of SCGO in obtaining an optimal solution with a very small number of function evaluations comparing to the commonly existing solvers, there are some other points regarding the SCGO algorithm as below:

- **Exploration:** The SCGO algorithm employs space-filling design sampling methods (e.g. LHS) to construct Kriging surrogate. This method spreads sample points in the whole design space. However, the search procedure is performed in the whole design space using the grid-search method. The proposed algorithm searches for the optimal solution by employing Kriging instead of an expensive original model using the "grid- search" method. In this approach, the whole design space is studied through the evaluation model's components consisting of objectives and constraints set in the equally spaced grid points. In this searching method, there is a chance to find the global optimum solution because searching is not bounded to the local valley (global search).
- **Exploitation:** In the proposed algorithm, the adaptive improvement is used to search in the region that the best feasible point is located (local search), see Algorithm 2.

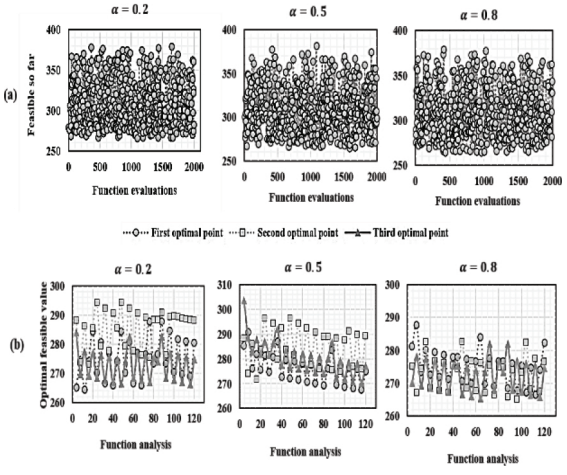

Figure 19 (a) Feasible points regarding sorted grid points (search sample points) and (b) adaptive improvement for three optimal points obtained from sorted grid points in three-bar truss design problem.

Table 4 Comparison of SCGO optimization results with literature for the three-bar truss design problem.

Algorithm		Optimum Variables X1	Optimum Variables X2	Optimal Weight	Function Evaluations
SCGO	$\alpha = 0.2$	0.80889	0.35538	264.32708	222
	$\alpha = 0.5$	0.74425	0.56917	267.42254	228
	$\alpha = 0.8$	0.76080	0.49944	265.12941	226
GOA		0.78890	0.40762	263.89588	13000
ALO		0.78866	0.40828	263.89584	14000
PSO-DE		0.78868	0.40825	263.89584	17600

SCGO, Surrogate-Based Constrained Global-Optimization; GOA, Grasshopper Optimization Algorithm; ALO, Ant Lion Optimizer; PSO-DE, Particle Swarm Optimization with Differential Evolution.

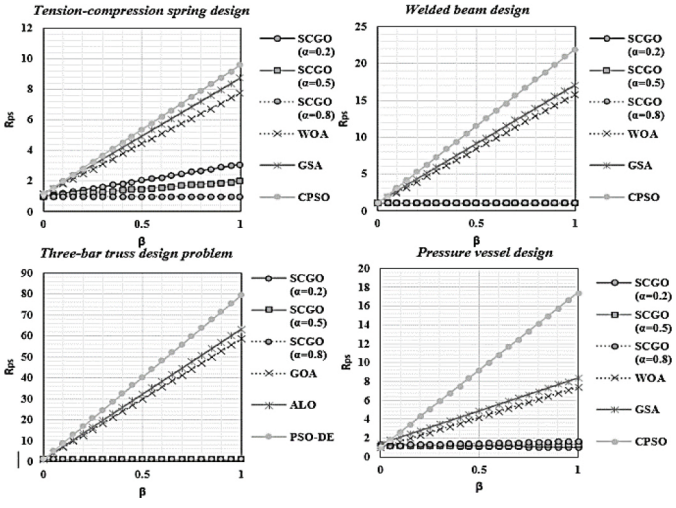

Figure 20 The performance comparison of Surrogate-Based Constrained Global-Optimization (SCGO) with other solvers in the literature for four common engineering design problems. The performance criterion $R_{p,s}$ measured based on two terms, accuracy of solution (lower objective function), and number of function evaluations (computational cost), see Eq. (13).

The SCGO algorithm also provides very competitive results on the benchmark engineering design problems. This demonstrates that SCGO shows a good balance between exploration and exploitation to the avoidance of local optima.

4.2. Limitations of SCGO

There are some limitations of the proposed SCGO method as follows:

- The SCGO method employs Kriging surrogate to train the overall loss function including integrated objective and constraint functions using space-filling sampling strategies. Therefore, the approximate errors cannot be ignored when solving inequality-constrained simulation-optimization problems particularly with complex function and nonlinear structure. It is well known that Kriging models are ideal candidates for smooth models. If the functions are nonsmooth or noisy, it is likely that Kriging surrogate degrades rapidly and overfit due to their interpolating behavior. A challenge for optimization under restricted budgets will be to find the right degree of approximation (smoothing factor) from only relatively few samples [33].
- This study aims to provide a fair comparison between the proposed algorithm and the highly referred methods in the literature [10–12,84,85] for solving benchmark problems. Therefore, the same framework has been taken from the mentioned studies (i.e., the problems with all inequality constraints). In [85] and [33], a reformulation of an equality constraint of $h(x) = 0$ to inequality constraint using $|h(x)| - \varepsilon \leq 0$ has performed to handle inequalities in the model, where ε is the tolerance allowed (a very small value). The same strategy can be used in this study. However, it must be noted that this equality-to-inequality transformation severely changes the nature of the optimization problem since it fundamentally changes the feasible volume. Regarding this point of view, it can be only said that the proposed SCGO approach in this study solved benchmarks with all inequality constraints well. Moreover, developing the SCGO in order to properly handle equality constraint is left to the future remarks.
- This study aims to provide a fair comparison between SCGO and the highly referred methods in the literature for solving problems in practical engineering design optimization. Therefore, the same framework (low-dimensional optimization problems) have been taken from studies [10–12,84,85]. However, in this study, the application of SCGO is limited to low-dimensional optimization problems.

5. CONCLUSION

In the paper, the SCGO algorithm is developed to solve simulation-based optimization problems with a computationally expensive objective and constraints. Kriging surrogate is used to approximate the computational expensive overall loss function. Firstly, an overall loss function is constructed to integrate objective and constraint functions in the same overall objective function. Secondly, the Kriging surrogate is constructed using a small number of simulation experiments. Then, the adaptive improvement approach is performed to further improve the optimal solution while enforcing a feasible solution. The grid-search strategy is used to effectively perform exploration in whole design space and the avoidance of local optima. The proposed algorithm is tested using several constrained global optimization problems. The results indicate that the SCGO method is effective for solving expensive constrained optimization problems with a much smaller number of function evaluations comparing to some state-of-the-art metaheuristics.

Future research will be devoted to overcoming the current limitations of SCGO mentioned in Section 4.2. It would be interesting to extend the proposed SGCO algorithm to handle high-dimensional-constrained optimization problems while keeping performance in obtaining optimal solutions with a small number of function evaluations. Besides, the SGCO algorithm can be extended to cover equality constraints as well as inequality constraint functions. Kriging can be replaced with other interpolation methods such as radial basis function, support vector machine, or artificial neural network. Besides, the application of SCGO can be developed to handle a higher dimension of optimization problems in the practice of constrained engineering design problems.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interests.

AUTHORS' CONTRIBUTIONS

All authors have contributed equally to this study. All authors read and approved the final manuscript.

ACKNOWLEDGMENTS

This research project is supported by Second Century Fund (C2F), Chulalongkorn University, Bangkok, Thailand.

REFERENCES

[1] O.A. Arqub, Z. Abo-Hammour, Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm, Inf. Sci. (Ny). 279 (2014), 396–415.
[2] H.G. Beyer, B. Sendhoff, Robust optimization - a comprehensive survey, Comput. Methods Appl. Mech. Eng. 196 (2007), 3190–3218.
[3] J. Stork, *et al.*, Open issues in surrogate-assisted optimization, in: T. Bartz-Beielstein, B. Filipič, P. Korošec, E.G. Talbi (Eds.), High-Performance Simulation-Based Optimization, Studies in Computational Intelligence, vol. 833, Springer, Cham, Switzerland, 2020, pp. 225–244.
[4] N. Bartoli, *et al.*, Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design, Aerosp. Sci. Technol. 90 (2019), 85–102.
[5] F.A.C. Viana, T.W. Simpson, V. Balabanov, V. Toropov, Metamodeling in multidisciplinary design optimization: how far have we really come?, AIAA J. 52 (2014), 670–690.

[6] A. Mohammad Nezhad, H. Mahlooji, An artificial neural network meta-model for constrained simulation optimization, J. Oper. Res. Soc. 65 (2013), 1232–1244.
[7] E. Bonabeau, D.R.D.F. Marco, M. Dorigo, G. Theraulaz, Swarm Intelligence: from Natural to Artificial Systems, Oxford University Press, New York, 1999.
[8] M. Dorigo, M. Birattari, T. Stutzle, Ant Colony Optimization, IEEE computational intelligence magazine, Boston, MA, USA, 1 (2006), 28–39.
[9] R. Eberhart, J. Kennedy, Particle swarm optimization, in Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1995, vol. 4, pp. 1942–1948.
[10] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014), 46–61.
[11] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016), 51–67.
[12] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015), 80–98.
[13] E. Bonabeau, Agent-based modeling: methods and techniques for simulating human systems, Proc. Natl. Acad. Sci. 99 (2002), 7280–7287.
[14] L.A. Zadeh, Fuzzy logic, Computer. 21 (1988), 83–93.
[15] O.A. Arqub, M. Al-Smadi, Fuzzy conformable fractional differential equations: novel extended approach and new numerical solutions, Soft Comput. 24 (2020), 12501–12522.
[16] D. Graupe, Principles of Artificial Neural Networks, World Scientific Publishing Company, Singapore, vol. 7 (2013).
[17] T.W. Simpson, J.D. Poplinski, P.N. Koch, J.K. Allen, Metamodels for computer-based engineering design: survey and recommendations, Eng. Comput. 17 (2001), 129–150.
[18] G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, J. Mech. Des. 129 (2007), 370–380.
[19] A. Parnianifard, A. Azfanizam, M. Ariffin, M. Ismail, N. Ebrahim, Recent developments in metamodel based robust black-box simulation optimization: an overview, Decis. Sci. Lett. 8 (2019), 17–44.
[20] A. Parnianifard, A.S. Azfanizam, M.K.A. Ariffin, M.I.S. Ismail, An overview on robust design hybrid metamodeling: advanced methodology in process optimization under uncertainty, Int. J. Ind. Eng. Comput. 9 (2018), 1–32.
[21] R. Yondo, E. Andrés, E. Valero, A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses, Prog. Aerosp. Sci. 96 (2018), 23–61.
[22] J. Xu, E. Huang, C.H. Chen, L.H. Lee, Simulation optimization: a review and exploration in the new era of cloud computing and big data, Asia-Pacific J. Oper. Res. 32 (2015), 1550019.
[23] Y.F. Li, S.H. Ng, M. Xie, T.N. Goh, A systematic comparison of metamodeling techniques for simulation optimization in decision support systems, Appl. Soft Comput. 10 (2010), 1257–1273.
[24] S. Jacobson, L. Schruben, A Review of Techniques for Simulation Optimization, Cornell University Operations Research and Industrial Engineering, New York, 1986, https://hdl.handle.net/1813/8598.
[25] S. Amaran, N.V. Sahinidis, B. Sharda, S.J. Bury, Simulation optimization: a review of algorithms and applications, Ann. Oper. Res. 240 (2016), 351–380.
[26] T. Bartz-Beielstein, B. Filipič, P. Korošec, E.-G. Talbi, High-Performance Simulation-Based Optimization, first ed., Springer Nature Switzerland AG, Cham Switzerland, 2020.
[27] B. Ali Asghar, Computational intelligence and its applications in uncertainty-based design optimization, in: M.A. Wahab, Y.L. Zhou (Eds.), Bridge Optimization-Inspection and Condition Monitoring, IntechOpen Limited, London, UK, 2019.
[28] A. Parnianifard, A.S. Azfanizam, M.K.A. Ariffin, M.I.S. Ismail, Comparative study of metamodeling and sampling design for expensive and semi-expensive simulation models under uncertainty, Simulation. 96 (2019), 89–110.
[29] A. Parnianifard, A. Azfanizam, Metamodel-based robust simulation-optimization assisted optimal design of multiloop integer and fractional-order PID controller, Int. J. Numer. Model. Electron. Netw. Devices Fields. 33 (2020), e2679.
[30] A. Parnianifard, A.S. Azfanizam, M.K.A. Ariffin, M.I.S. Ismail, Kriging-assisted robust black-box simulation optimization in direct speed control of DC motor under uncertainty, IEEE Trans. Magn. 54 (2018), 1–10.
[31] R. Jin, X. Du, W. Chen, The use of metamodeling techniques for optimization under uncertainty, Struct. Multidiscipl. Optim. 25 (2003), 99–116.
[32] Y. Wu, Q. Yin, H. Jie, B. Wang, J. Zhao, A RBF-based constrained global optimization algorithm for problems with computationally expensive objective and constraints, Struct. Multidiscip. Optim. 58 (2018), 1633–1655.
[33] S. Bagheri, W. Konen, M. Emmerich, T. Bäck, Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets, Appl. Soft Comput. J. 61 (2017), 377–393.
[34] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (2009), 2232–2248.
[35] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017), 163–191.
[36] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (2007), 89–99.
[37] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Softw. 105 (2017), 30–47.
[38] C.A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Eng. 191 (2002), 1245–1287.
[39] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[40] Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, Evol. Comput. 4 (1996), 1–32.
[41] L. Jiao, L. Li, R. Shang, F. Liu, R. Stolkin, A novel selection evolutionary strategy for constrained optimization, Inf. Sci. 239 (2013), 122–141.
[42] D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions, J. Glob. Optim. 13 (1998), 455–492.
[43] L. Peng, L. Liu, T. Long, X. Guo, Sequential RBF surrogate-based efficient optimization method for engineering design problems with expensive black-box functions, Chinese J. Mech. Eng. 27 (2014), 1099–1111.

[44] J.P.C. Kleijnen, W. Van Beers, I. Van Nieuwenhuyse, Expected improvement in efficient global optimization through bootstrapped kriging, J. Glob. Optim. 54 (2012), 59–73.
[45] E. Mehdad, J.P.C. Kleijnen, Efficient global optimisation for black-box simulation via sequential intrinsic Kriging, J. Oper. Res. Soc. 5682 (2018), 1–13.
[46] D.R. Jones, A taxonomy of global optimization methods based on response surfaces, J. Glob. Optim. 21 (2001), 345–383.
[47] M.J. Sasena, P. Papalambros, P. Goovaerts, Exploration of metamodeling sampling criteria for constrained global optimization, Eng. Optim. 34 (2002), 263–278.
[48] D. Huang, T.T. Allen, W.I. Notz, N. Zeng, Global optimization of stochastic black-box systems via sequential kriging meta-models, J. Glob. Optim. 34 (2006), 441–466.
[49] Y. Tang, J. Chen, J. Wei, A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions, Eng. Optim. 45 (2013), 557–576.
[50] R.G. Regis, Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions, IEEE Trans. Evol. Comput. 18 (2013), 326–347.
[51] C.M.C. Razali, S.S. Abdullah, A. Parnianifard, A. Faruq, Adaptive infill sampling strategy for metamodeling challenge and future research directions, Bull. Electr. Eng. Informat. 9 (2020), 2020–2029.
[52] C. Audet, J. Denni, D. Moore, A. Booker, P. Frank, A surrogate-model-based method for constrained optimization, in 8th Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, USA, 2000, p. 4891.
[53] C. Durantin, J. Marzat, M. Balesdent, Analysis of multi-objective Kriging-based methods for constrained global optimization, Comput. Optim. Appl. 63 (2016), 903–926.
[54] Z. Wang, M. Ierapetritou, Constrained optimization of black-box stochastic systems using a novel feasibility enhanced Kriging-based method, Comput. Chem. Eng. 118 (2018), 210–223.
[55] S. Gano, H. Kim, D. Brown, Comparison of three surrogate modeling techniques: datascape, kriging, and second order regression, in Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (AIAA-2006–7048), Portsmouth, VA, USA, 2006, 3:
[56] R. Jin, W. Chen, T.W. Simpson, Comparative studies of metamodelling techniques under multiple modelling criteria, Struct. Multidiscip. Optim. 23 (2001), 1–13.
[57] P. Koch, S. Bagheri, W. Konen, C. Foussette, P. Krause, T. Bäck, A new repair method for constrained optimization, in Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, Madrid, Spain, 2015, pp. 273–280.
[58] P. Koch, S. Bagheri, C. Foussette, P. Krause, T. Bäck, W. Konen, Constrained optimization with a limited number of function evaluations, in ProcEEDings 24, Workshop comPutational intElligEncE, Dortmund, Germany, 2014, p. 237.
[59] R.G. Regis, Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points, Eng. Optim. 46 (2014), 218–243.
[60] H. Xiao, W. Pei, Z. Dong, L. Kong, D. Wang, Application and comparison of metaheuristic and new metamodel based global optimization methods to the optimal operation of active distribution networks, Energies. 11 (2018), 85.
[61] S.S. Garud, I.A. Karimi, M. Kraft, Design of computer experiments: a review, Comput. Chem. Eng. 106 (2017), 71–95.
[62] J.P.C.C. Kleijnen, Design and Analysis of Simulation Experiments, second ed., Springer, Cham, Springer International Publishing Switzerland, 2015.
[63] A. Parnianifard, A.S. Azfanizam, M.K.A. Ariffin, M.I.S. Ismail, Trade-off in robustness, cost and performance by a multi-objective robust production optimization method, Int. J. Ind. Eng. Comput. 10 (2019), 133–148.
[64] M.S. Phadke, Quality Engineering Using Robust Design, Prentice Hall PTR, Springer, Boston, MA, 1989.
[65] V.R. Joseph, Space-filling designs for computer experiments: a review, Qual. Eng. 28 (2016), 28–35.
[66] T. Bartz-Beielstein, C. Jung, M. Zaefferer, Uncertainty management using sequential parameter optimization, in: G. Dellino, C. Meloni (Eds.), Uncertainty Management in Simulation-Optimization of Complex Systems, vol. 59, Springer, Boston, MA, USA, 2015.
[67] E. Del Castillo, Process Optimization A Statistical Approach, vol. 480, Springer, Boston, MA, USA, 2007.
[68] J.R. Koehler, A.B. Owen, Computer experiments, Handbook Stat. 13 (1996), 261–308.
[69] T.W. Simpson, T.M. Mauery, J. Korte, F. Mistree, Kriging models for global approximation in simulation-based multidisciplinary design optimization, AIAA J. 39 (2001), 2233–2241.
[70] J.P.C. Kleijnen, Kriging metamodeling in simulation: a review, Eur. J. Oper. Res. 192 (2009), 707–716.
[71] S.N. Lophaven, J. Søndergaard, H.B. Nielsen, DACE a Matlab Kriging Toolbox, Informatics and Mathematical Modelling, Technical University of Denmark, DTU,2002, pp. 1–28, http://www2.imm.dtu.dk/pubdb/p.php?1460.
[72] F. Jurecka, Robust Design Optimization Based on Metamodeling Techniques, PhD Thesis, Technische Universität München, München, Germany, 2007, https://mediatum.ub.tum.de/doc/607314/446546.pdf.
[73] R. Myers, D.C. Montgomery, C. Anderson-Cook, Response Surface Methodology: Process and Product Optimization Using Designed Experiments, fourth ed., John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 2016.
[74] K. Rutten, Methods For Online Sequential Process Improvement, PhD Thesis, KU Leuven univeristy, 3000 Leuven, Belgium, 2015,https://lirias.kuleuven.be/1715807?limo=0.
[75] M.K. Zahir, Z. Gao, Variable-fidelity optimization with design space reduction, Chinese J. Aeronaut. 26 (2013), 841–849.
[76] S. Shan, G.G. Wang, Space exploration and global optimization for computationally intensive design problems: a rough set based approach, Struct. Multidiscip. Optim. 28 (2004), 427–441.
[77] V.V. de Melo, A.C.B. Delbem, D.L. Pinto Junior, F.M. Federson, Improving global numerical optimization using a search-space reduction algorithm, in Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, England, 2007, pp. 1195–1202.
[78] J.S. Arora, Introduction to Optimum Design, Elsevier, Academic Press, Cambridge, Massachusetts, United States, 2004.
[79] C.A. Coello Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2000), 113–127.
[80] B.K. Kannan, S.N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, J. Mech. Des. Trans. ASME. 116 (1994), 318–320.

[81] H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, Appl. Soft Comput. 10 (2010), 629–640.
[82] T.P. Runarsson, X. Yao, Stochastic ranking for constrained evolutionary optimization, IEEE Trans. Evol. Comput. 4 (2000), 284–294.
[83] J.J. Moré, S.M. Wild, Benchmarking derivative-free optimization algorithms, SIAM J. Optim. 20 (2009), 172–191.
[84] S. Mirjalili, S.M. Mirjalili, X.S. Yang, Binary bat algorithm, Neural Comput. Appl. 25 (2014), 663–681.
[85] C.A. Coello Coello, E.M. Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Adv. Eng. Informat. 16 (2002), 193–203.