Taylor & Francis
Taylor & Francis Group

Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: https://www.tandfonline.com/loi/geno20

A parallel constrained efficient global optimization algorithm for expensive constrained optimization problems

Jiachang Qian, Yuansheng Cheng, Jinlan Zhang, Jun Liu & Dawei Zhan

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2020.1722118

A parallel constrained efficient global optimization algorithm for expensive constrained optimization problems

Jiachang Qian[a,b], Yuansheng Cheng[a], Jinlan Zhang[b], Jun Liu[a] and Dawei Zhan [a,c]

[a]School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan, PR China; [b]Wuhan Second Ship Design and Research Institute, Wuhan, PR China; [c]School of Information Science and Technology, Southwest Jiaotong University, Chengdu, PR China

ABSTRACT

The Constrained Expected Improvement (CEI) criterion used in the so-called Constrained Efficient Global Optimization (C-EGO) algorithm is one of the most famous infill criteria for expensive constrained optimization problems. However, the standard CEI criterion selects only one point to evaluate in each cycle, which is time consuming when parallel computing architecture is available. This work proposes a new Parallel Constrained EGO (PC-EGO) algorithm to extend the C-EGO algorithm to parallel computing. The proposed PC-EGO algorithm is tested on sixteen analytical problems as well as one real-world engineering problem. The experiment results show that the proposed PC-EGO algorithm converges significantly faster and finds better solutions on the test problems compared to the standard C-EGO algorithm. Moreover, when compared to another state-of-the-art parallel constrained EGO algorithm, the proposed PC-EGO algorithm shows more efficient and robust performance.

ARTICLE HISTORY
Received 30 November 2017
Accepted 23 January 2020

KEYWORDS
Efficient global optimization; surrogate model; parallel computing; expensive optimization; constrained optimization

1. Introduction

The analysis of a real-world engineering system often involves computationally expensive simulations, and the optimization design of these expensive black-box problems has become a popular research topic in recently years (Jones, Schonlau, and Welch 1998; Wang, Shan, and Wang 2004; Regis and Shoemaker 2005; Rashid, Ambani, and Cetinkaya 2013). Since the objectives and constraints of these problems are calculated using computer simulations, the derivatives cannot be obtained directly, which prohibits the use of derivative-based algorithms. Nature-inspired algorithms such as the Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO) algorithm have been widely used to solve complex engineering optimization problems. However, these population-based optimization algorithms often require tens or hundreds of thousands evaluations of the problems they face, which makes them unsuitable for expensive optimization problems. Using surrogates in lieu of expensive simulations within the optimization process has become a popular method to solve expensive black-box optimization problems (Simpson *et al.* 2004; Wang and Shan 2007; Forrester and Keane 2009; Younis and Dong 2010; Viana *et al.* 2014). In these surrogate-based optimization algorithms, one or more surrogates are built based on a small set of evaluated points to approximate the expensive objectives and constraints. The algorithms try to find the global optima of the original problems using as few evaluations of the expensive simulations as possible.

CONTACT Dawei Zhan zhandawei@swjtu.edu.cn

Surrogates are used in different ways among different surrogate-based methods. There are approaches that couple surrogate models with evolutionary algorithms (Jin, Olhofer, and Sendhoff 2002; Shahrokhi and Jahangirian 2010; Jin 2011; Tang, Chen, and Wei 2013; Saad *et al.* 2019). These algorithms are often called surrogate-assisted evolutionary algorithms. Surrogate models are treated as fast fitness calculators and are used to identify promising individuals from candidates to do expensive simulations. The search for the global optimum is still done through the evolution of the populations in these algorithms. Another kind of approach tries to find the global optimum through sequentially selecting new design points, evaluating the selected points and updating the surrogate model. Efficient Global Optimization (EGO) (Schonlau 1997; Jones, Schonlau, and Welch 1998), the Adaptive Response Surface Method (ARSM) (Wang, Dong, and Aitchison 2001; Wang 2003), the Mode Pursing Method (MPS) (Wang, Shan, and Wang 2004; Duan *et al.* 2009) and algorithms like those in Regis and Shoemaker (2013), Müller and Shoemaker (2014), Jie, Wu, and Ding (2015), Long *et al.* (2015), Müller and Woodbury (2017) and Liu *et al.* (2017) belong to this kind of approach. The infill sampling criterion, based on which the updating points are selected, is the key factor to the efficiencies of these sequential sampling approaches. The infill sampling criterion should on the one hand select points near the current best observation to exploit the most promising areas, and on the other hand select points far away from current observations to explore the undiscovered areas. A balance between local search and global search is needed in order to search for the global optimum effectively in these sequential sampling approaches.

Many real-world optimization problems often have both expensive objective function and expensive constraint functions. In order to deal with these expensive constrained optimization problems, Kazemi *et al.* (2011) extended the MPS algorithm to deal with expensive constraints. Regis (2011) proposed the Constrained Local Metric Stochastic Radial Basis Function (ConstrLMSRBF) algorithm, which uses RBF models to approximate the expensive objective and constraints. Then Regis (2014) further extended his algorithm and proposed the Constrained Optimization By RAdial basis function interpolation (COBRA) algorithm with the ability to solve expensive constrained optimization problems when all initial design points are infeasible. Recently, Shi *et al.* (2018) proposed the Filter-based Sequential Radial Basis Function (FSRBF) method to solve expensive constrained optimization problems based on RBF models and a filter mechanism.

Among different surrogate-based optimization algorithms, the EGO algorithm (Schonlau 1997; Jones, Schonlau, and Welch 1998) is one of the most famous and widely studied algorithms. It uses the Kriging model to approximate the expensive objective function, and uses the so-called Expected Improvement (EI) criterion to select an updating point in each cycle. However, the standard EI criterion did not consider computationally expensive constraints, which often occur in real-world optimization problems. In order to deal with expensive constraints, Schonlau (1997) proposed to use the Probability of Feasibility (PoF). The PoF criterion measures the probability that an unknown point will satisfy the constraints. Audet *et al.* (2000) proposed to use the Expected Violation (EV) criterion to measure how much a candidate will violate the constraints, and only to use selected points that have smaller EV value than a tolerance. Sasena, Papalambros, and Goovaerts (2002) used a penalty method to restrict the sampling criterion from choosing points in infeasible space. Gramacy *et al.* (2016) used the augmented Lagrangian method to turn the constrained optimization problem into an unconstrained one, and derived a new EI criterion for the augmented Lagrangian function. Among these constraint handling methods, the PoF function is still one of the most widely used criteria because of its simplicity and efficiency. When dealing with computationally expensive constraints, the PoF function is simply multiplied by the EI function to form the Constrained EI (CEI) criterion. By maximizing the CEI criterion, updating points can be selected that not only have great improvements in the objective function but also have high probability of satisfying the constraints.

However, the standard CEI criterion is a sequential criterion, it can select only one updating point to evaluate in each optimization cycle. When parallel computing architecture is available, the standard CEI criterion cannot take full advantage of parallel computing. Therefore, it is urgent to extend the sequential (one-point) CEI criterion into a parallel (multi-point) criterion. Most ideas and approaches in the literature aim to develop a parallel unconstrained EI criterion, few focus on developing a parallel constrained EI criterion. Sóbester, Leary, and Keane (2004) proposed a method that selects multiple updating points at the different maxima of the EI function. Ginsbourger, Riche, and Carraro (2010) worked on the q-EI criterion, which measures the expected improvement value when q $(q \geq 2)$ updating points are added to the design set. The formula was derived for $q = 2$ and Monte Carlo sampling was recommended for $q > 2$ cases. Viana, Haftka, and Watson (2013) built multiple surrogate models and multiple EI functions, and selected one updating point from one EI function. The number of updating points in their approach is restricted by the number of surrogate models that are used. Feng *et al.* (2015) treated the EI function as two parts: a local search part and global search part. A bi-objective optimization was used to maximize these two parts at the same time, then multiple updating points were selected from the obtained Pareto fronts. How to select multiple promising updating points from a large number of Pareto front points is still an open question. Recently, a parameter-less parallel EI criterion called the pseudo EI criterion has been proposed by the authors of this article (Zhan, Qian, and Cheng 2017), which uses an influence function to simulate the effect that the updating point will bring to the EI function. The pseudo EI function was used to approximate the EI function in the following stages to produce multiple candidates within one optimization cycle.

In terms of a parallel constrained EI criterion, Parr *et al.* (2012) proposed a bi-objective approach. The bi-objective approach maximizes the EI function and the PoF function at the same time, and selects multiple candidates from the obtained Pareto optimal points. The idea of the bi-objective approach is very natural. Since candidates with both high EI value and PoF value are wanted, it is reasonable to treat the candidate selecting problem as a bi-objective problem and solve it with a bi-objective optimization algorithm. The problem of the bi-objective approach is that the number of obtained Pareto front points is often significantly more than the number of selected updating points. How to identify the most promising candidates from a large number of Pareto set points can still be a difficult problem.

In this article, a new parallel CEI criterion is proposed. The idea of the influence function (Zhan, Qian, and Cheng 2017) is used to extend the CEI criterion to produce multiple candidates in each cycle. The proposed criterion is very cheap to calculate and easy to implement. In order to validate the efficiency of the proposed criterion, it is compared to the standard CEI criterion as well as the parallel bi-objective CEI criterion on sixteen analytical problems and one real-world engineering problem.

2. The standard constrained EGO algorithm

The standard Efficient Global Optimization (EGO) algorithm was popularized by Jones, Schonlau, and Welch (1998) and extended to handle expensive constraints by Schonlau (1997). The Constrained EGO (C-EGO) algorithm solves an optimization problem with one expensive objective and several expensive constraints. In the C-EGO algorithm, the objective and constraints are all approximated by Kriging models.

2.1. Kriging model

The Kriging model was first used in geology to approximate the distribution of minerals, and was introduced to approximate computer experiments by Sacks *et al.* (1989) under the name of Design and Analysis of Computer Experiments (DACE). Given a set of design points $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}^{\mathrm{T}}$ and their outputs $\mathbf{y} = \{y^{(1)}, y^{(2)}, \ldots, y^{(n)}\}^{\mathrm{T}}$, the prediction as well as the variance of the prediction of any design point $\mathbf{x}$ can be derived as (Sacks *et al.* 1989)

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) \tag{1}$$

and

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

In the above two equations, $\hat{\mu}$ and $\hat{\sigma}^2$ are the maximal likelihood estimations of the mean and variance of the Gaussian process, respectively, $\mathbf{r}$ is an n-dimensional vector with element $r_i = \mathrm{Corr}(\mathbf{x}, \mathbf{x}^{(i)})$ where $\mathrm{Corr}(\cdot)$ is the correlation function defined in the Kriging model, $\mathbf{R}$ is an $n \times n$ matrix with entry $R_{ij} = \mathrm{Corr}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ and $\mathbf{1}$ is an n-dimensional vector of ones.

2.2. The constrained expected improvement criterion

For an unknown point $\mathbf{x}$, its objective and constraint values can be seen as random values

$$Y(\mathbf{x}) \sim N(\hat{y}(\mathbf{x}), s^2(\mathbf{x})) \tag{3}$$

and

$$G_i(\mathbf{x}) \sim N(\hat{g}_i(\mathbf{x}), s_i^2(\mathbf{x})), \quad i = 1, 2, \ldots, c, \tag{4}$$

where $\hat{y}$ and s are the Kriging prediction and standard error of the objective function, and $\hat{g}_i$ and s_i are the Kriging prediction and standard error of the ith constraint function. The Constrained EI (CEI) is the measurement of improvement that the unknown point can achieve when it satisfies all the constraints (Schonlau 1997):

$$\begin{aligned}\mathrm{CEI}(\mathbf{x}) &= \mathrm{EI}(\mathbf{x}) \times \mathrm{PoF}(\mathbf{x}) \\ &= \left[(y_{\min} - \hat{y}(\mathbf{x}))\Phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right)\right] \\ &\quad \times \prod_{i=1}^{c} \Phi\left(\frac{-\hat{g}_i(\mathbf{x})}{s_i(\mathbf{x})}\right),\end{aligned} \tag{5}$$

where $\mathrm{EI}(\cdot)$ is the unconstrained Expected Improvement function, $\mathrm{PoF}(\cdot)$ is the Probability of Feasibility measurement, $y_{\min}$ is the current minimum feasible objective value, and $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative density function and probability density function of the normal distribution, respectively.

2.3. The process of the standard constrained EGO algorithm

The C-EGO algorithm is a two-step algorithm. In the first step of the algorithm, the initial Kriging models of the objective and constraints are built based on a small set of evaluated design points. Then, in the second step, the algorithm iteratively selects one point to evaluate until the stopping condition is met. When there is no feasible solution found in the current evaluated points, the C-EGO algorithm uses the PoF criterion to select potentially feasible points. Once one feasible point has been identified, the C-EGO algorithm shifts to the CEI criterion to improve the objective.

3. The proposed parallel constrained EGO algorithm

One of the major problems with the standard C-EGO algorithm is that it selects only one updating point to evaluate in each iteration, thus cannot utilize parallel computing technology. In present-day engineering industry, computational resources are increasing rapidly, and parallel computing architecture is often available. In order to apply the C-EGO algorithm to parallel computing, the CEI criterion is extended in this work to produce multiple candidates by using the idea of the influence function (Zhan, Qian, and Cheng 2017).

3.1. The influence function

The basic idea of the influence function is to use an artificial function to approximate the effect that the updating point will bring to the CEI function. Then the first CEI function can be multiplied by the influence function to approximate the second CEI function. So instead of calculating the real function value of the first updating point to update the Kriging model to get the second updating point, the approximated second CEI function can be used for selecting the second updating point. The key point is that the influence functions of the updating points are only determined by their design variables, not by their function values. As a result, multiple updating points can be selected without evaluating the previous points. The Influence Function (IF) of an updating point $\mathbf{x}^u$ proposed in Zhan, Qian, and Cheng (2017) is

$$\mathrm{IF}(\mathbf{x}, \mathbf{x}^u) = 1 - \mathrm{Corr}\big(\mathbf{x}, \mathbf{x}^u\big). \tag{6}$$

3.2. The proposed parallel constrained EGO (PC-EGO) algorithm

Since the influence function is a continuous function with value between zero and one, it can be coupled perfectly with the CEI criterion. By simply multiplying the the CEI function by the IF function, the proposed Pseudo Constrained EI (PCEI) criterion can be calculated as

$$\mathrm{PCEI}(\mathbf{x}, \mathbf{x}^{(u)}) = \mathrm{CEI}(\mathbf{x}) \times \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(u)}). \tag{7}$$

Based on the PCEI criterion, q updating points can easily be picked as follows. The first updating point is selected by the standard CEI function

$$\mathbf{x}^{(n+1)} = \mathrm{argmax}\ \mathrm{CEI}(\mathbf{x}). \tag{8}$$

Then the second updating point can be selected by

$$\mathbf{x}^{(n+2)} = \mathrm{argmax}\ \mathrm{CEI}(\mathbf{x}) \times \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+1)}). \tag{9}$$

And as the process goes on, the qth updating point can be selected as

$$\mathbf{x}^{(n+q)} = \mathrm{argmax}\ \mathrm{CEI}(\mathbf{x}) \times \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+1)}) \times \cdots \times \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+q-1)}). \tag{10}$$

Substituting the $\mathrm{Corr}(\cdot)$ and $\mathrm{CEI}(\cdot)$ functions into this equation, the close-form expression of the PCEI function can be derived:

$$\begin{aligned}
&\mathrm{PCEI}(\mathbf{x}, \mathbf{x}^{(n+1)}, \ldots, \mathbf{x}^{(n+q-1)}) \\
&\quad = \mathrm{CEI}(\mathbf{x}) \times \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+1)}) \times \cdots \times \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+q-1)}) \\
&\quad = \left[\big(y_{\min} - \hat{y}(\mathbf{x})\big) \Phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x}) \phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right)\right] \prod_{i=1}^{c} \Phi\left(\frac{-\hat{g}_i(\mathbf{x})}{s_i(\mathbf{x})}\right) \\
&\qquad \times \prod_{i=1}^{q-1} \left[1 - \exp\left(-\sum_{k=1}^{d} \theta_k \left|\mathbf{x}_k - \mathbf{x}_k^{(n+i)}\right|^{p_k}\right)\right],
\end{aligned} \tag{11}$$

where $y_{\min}$ is the current best feasible objective value, $\hat{y}(\mathbf{x})$ and $s(\mathbf{x})$ are the prediction and standard deviation of the objective Kriging model, $\hat{g}_i(\mathbf{x})$ and $s_i(\mathbf{x})$ $(i = 1, 2, \ldots, c)$ are the prediction and standard deviation of the constraint Kriging model, and θ_k and p_k $(k = 1, 2, \ldots, d)$ are parameters of the objective Kriging model, which are determined after the Kriging model is built.

Additionally, the influence function can also be applied to the PoF criterion to select multiple updating points when there is no feasible solution in the initial designs. The Pseudo PoF (PPoF) criterion can be expressed as

$$\begin{aligned}
&\mathrm{PPoF}(\mathbf{x}, \mathbf{x}^{(n+1)}, \ldots, \mathbf{x}^{(n+q-1)}) \\
&\quad = \mathrm{PoF}(\mathbf{x}) \times \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+1)}) \times \cdots \times \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+q-1)}) \\
&\quad = \prod_{i=1}^{c} \Phi\left(\frac{-\hat{g}_i(\mathbf{x})}{s_i(\mathbf{x})}\right) \times \prod_{i=1}^{q-1}\left[1 - \exp\left(-\sum_{k=1}^{d} \theta_k |\mathbf{x}_k - \mathbf{x}_k^{(n+i)}|^{p_k}\right)\right].
\end{aligned} \tag{12}$$

Based on the proposed PCEI criterion and the PPoF criterion, a new Parallel Constrained EGO (PC-EGO) algorithm is proposed. The procedure of the proposed PC-EGO algorithm can be summarized as Algorithm 1. The PC-EGO algorithm uses Kriging models to approximate the objective and constraints of the original problem. If there is no feasible solution in the current designs, the PPoF criterion is maximized to produce q updating points until at least one feasible solution is found. Then the infill criterion is switched to the PCEI criterion to produce q updating points to improve the current best feasible solution.

Algorithm 1 The framework of the proposed PC-EGO algorithm

Require: Initial design set $(\mathbf{X}, \mathbf{y}, \mathbf{g}_1, \ldots, \mathbf{g}_c)$
Ensure: The best feasible solution $(\mathbf{x}_{\min}, y_{\min})$
while the stop condition is not met **do**
  Build a Kriging model based on the current design set $(\mathbf{X}, \mathbf{y})$
  for $i = 1$ to c **do**
    Build a Kriging model based on the current design set $(\mathbf{X}, \mathbf{g}_i)$
  end for
  if there is no feasible solution found **then**
    for $i = 1$ to q **do**
      $\mathbf{x}^{(i)} = \operatorname{argmax} \mathrm{PPoF}(\mathbf{x}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(i-1)})$.
    end for
  else
    for $i = 1$ to q **do**
      $\mathbf{x}^{(i)} = \operatorname{argmax} \mathrm{PCEI}(\mathbf{x}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(i-1)})$.
    end for
  end if
  Evaluating $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(q)}\}$ in parallel with the real objective and constraints
  $\mathbf{X} \leftarrow \mathbf{X} \cup \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(q)}\}$
  $\mathbf{y} \leftarrow \mathbf{y} \cup \{y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(q)})\}$
  for $i = 1$ to c **do**
    $\mathbf{g}_i \leftarrow \mathbf{g}_i \cup \{g_i(\mathbf{x}^{(1)}), \ldots, g_i(\mathbf{x}^{(q)})\}$
  end for
  if there is no feasible solution found **then**
    print out ‘no feasible solution is found’
  else
    update the best feasible solution $(\mathbf{x}_{\min}, y_{\min})$
  end if
end while

3.3. Convergence analysis of the proposed PC-EGO algorithm

The convergence of the proposed PC-EGO algorithm can be stated as follows: if the number of possible sampling points is infinite, the PC-EGO algorithm based on the PPoF and PCEI criteria will visit all the sampling points and hence will always find the global feasible optimum. The convergence property is proved as follows.

First, the value of the PPoF criterion is analysed. The PPoF function can be rewritten as

$$\mathrm{PPoF}(\mathbf{x}, \mathbf{x}^{(n+1)}, \ldots, \mathbf{x}^{(n+q-1)}) = \mathrm{PoF}(\mathbf{x}) \times \prod_{i=1}^{q-1} \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+i-1)}). \tag{13}$$

It should be noted that the PPoF criterion is used to select points only when there is no feasible solution in the current design set. As a result, when the PPoF criterion is used, the sampled points $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}^{\mathbf{T}}$ are all infeasible. For the PoF function, it is easy to get $\mathrm{PoF}(\mathbf{x}) = 0$ at infeasible sampled points $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}^{\mathbf{T}}$, and $\mathrm{PoF}(\mathbf{x}) > 0$ at all the other points. For the IF function, from the expression in Equation (11), it is easy to get $\prod_{i=1}^{q-1} \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+i-1)}) = 0$ at points $\{\mathbf{x}^{(n+1)}, \mathbf{x}^{(n+2)}, \ldots, \mathbf{x}^{(n+q-1)}\}^{\mathbf{T}}$, and $\prod_{i=1}^{q-1} \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+i-1)}) > 0$ at all the other points. Then the values of PPoF in different cases can be derived as follows.

(1) At points $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}^{\mathbf{T}}$, $\mathrm{PoF} = 0$, thus PPoF is zero.
(2) At points $\{\mathbf{x}^{(n+1)}, \mathbf{x}^{(n+2)}, \ldots, \mathbf{x}^{(n+q-1)}\}^{\mathbf{T}}$, $\prod_{i=1}^{q-1} \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+i-1)}) = 0$, thus PPoF is zero.
(3) At points other than $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n+q-1)}\}^{\mathbf{T}}$, $\mathrm{PoF}(\mathbf{x}) > 0$, $\prod_{i=1}^{q-1} \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+i-1)}) > 0$, thus PPoF is positive.

Then the value of the PCEI criterion is analysed as follows. The PCEI function can be rewritten as

$$\mathrm{PCEI}(\mathbf{x}, \mathbf{x}^{(n+1)}, \ldots, \mathbf{x}^{(n+q-1)}) = \mathrm{EI}(\mathbf{x}) \times \mathrm{PoF}(\mathbf{x}) \times \prod_{i=1}^{q-1} \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+i-1)}). \tag{14}$$

Note that the PCEI criterion is used only when at least one sampled point is feasible. It is assumed that the sampled points $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k)}\}^{\mathbf{T}}$ $(k < n)$ are infeasible and $\{\mathbf{x}^{(k+1)}, \mathbf{x}^{(k+2)}, \ldots, \mathbf{x}^{(n)}\}^{\mathbf{T}}$ are feasible. For the infeasible sampled points, the PoF values are zeros. For the feasible sampled points, the EI values are zeros (since the EI value is the expected improvement beyond the best feasible sampled point). Then the PCEI value in different cases can be derived as follows.

(1) At points $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k)}\}^{\mathbf{T}}$, $\mathrm{PoF}(\mathbf{x}) = 0$, thus PCEI is zero.
(2) At points $\{\mathbf{x}^{(k+1)}, \mathbf{x}^{(k+2)}, \ldots, \mathbf{x}^{(n)}\}^{\mathbf{T}}$, $\mathrm{EI}(\mathbf{x}) = 0$, thus PCEI is zero.
(3) At points $\{\mathbf{x}^{(n+1)}, \mathbf{x}^{(n+2)}, \ldots, \mathbf{x}^{(n+q-1)}\}^{\mathbf{T}}$, $\prod_{i=1}^{q-1} \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+i-1)}) = 0$, thus PCEI is zero.
(4) At points other than $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n+q-1)}\}^{\mathbf{T}}$, $\mathrm{EI}(\mathbf{x}) > 0$, $\mathrm{PoF}(\mathbf{x}) > 0$, $\prod_{i=1}^{q-1} \mathrm{IF}(\mathbf{x}, \mathbf{x}^{(n+i-1)}) > 0$, thus PCEI is positive.

To sum up, the value of PPoF (or PCEI) is zero at points $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n+q-1)}\}^{\mathbf{T}}$ and is positive at all the other points. Then, as long as there are un-sampled points, the PPoF (or PCEI) criterion will never select a previously sampled or selected point. When the number of sampling points is assumed to be infinite, the PPoF (or PCEI) criterion will sample all points and the convergence to the global optimum can be guaranteed.

3.4. Search capability of the PCEI criterion

In order to study the search capability of the proposed PCEI criterion, the proposed PC-EGO algorithm is run on the Branin function (Forrester, Sóbester, and Keane 2008) for two iterations.

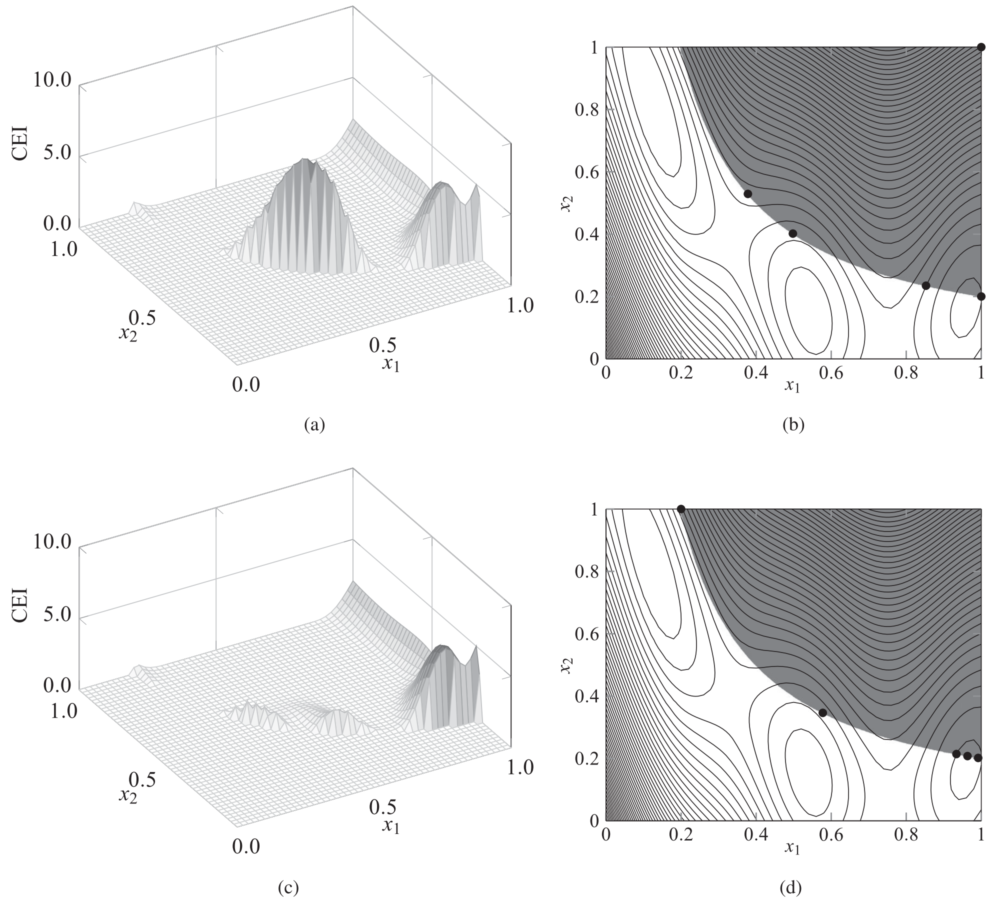


Figure 1. The first two iterations of the PC-EGO algorithm on the Branin function. (a) The CEI function of the first iteration. (b) The five updating points of the first iteration. (c) The CEI function of the second iteration and (d) The five updating points of the second iteration.

In each iteration, five updating points are selected according to the PCEI criterion. Figure 1 shows the CEI functions in each iteration and the corresponding five updating points selected by the PCEI criterion. In Figure 1(b,d), the black dots represents the five updating points selected by the PCEI criterion.

In the first iteration, the landscape of the CEI function is multi-modal. As a result, the PCEI criterion selects updating points in different CEI maxima region. The standard CEI criterion selects only one updating point at the maximum of the CEI function in one iteration. As a comparison, the proposed PCEI criterion is able to select multiple updating points around different CEI maxima. Different CEI maxima often indicate different potential improvements. By sampling multiple points at different CEI maxima, the PCEI criterion is able to search different areas at the same time, thus being able to speed up the search process. In the second iteration, the CEI function has a very high maximum and few small maxima. The PCEI criterion selects three updating points around the highest maximum and two points at smaller CEI maxima.

The distribution patterns of the two iterations are different. In the first iteration, the updating points are widely spread in the feasible space. In the second iteration, the updating points are more focused at the low right area where the true global feasible solution lies. At the beginning of the searching process, the CEI function often has many maxima. As a result, the updating points selected by PCEI criterion are spread out at different areas, exploring for potential areas. As the process goes on, the number of CEI maxima will become less and less. As a result, the updating points selected by the PCEI criterion will become more and more focused on certain regions, searching for the global optimum. In other words, the search pattern of the PCEI criterion will gradually change from global search to local search during the searching process of PC-EGO algorithm.

4. Numerical experiments

4.1. Test problems

Sixteen test problems are selected for the numerical experiments. Seven of the test problems are as follows.

(1) Branin function with a complex constraint (Branin) (Parr *et al.* 2012),
(2) Two-Member Frame (TMF) problem (Arora 1989),
(3) Gas Transmission Compressor Design (GTCD) problem (Beightler and Phillips 1976),
(4) Pressure Vessel Design (PVD) problem (Regis 2014),
(5) Welded Beam Design (WBD) problem (Rao 1996),
(6) Speed Reducer Design (SRD) problem (Floudas and Pardalos 1990), and
(7) Hesse problem (Hesse 1973).

The other nine test problems are from Michalewicz and Schoenauer (1996), Liang *et al.* (2006) and Regis (2014). They are labelled as G01, G02, G07, G08, G09, G13MOD, G16, G19 and G24. Detailed information about the test problems is given in Table 1. In the table, d represents the number of design variables and c represents the number of constraints.

4.2. Experiment setups

In order to study the efficiency of the proposed PC-EGO algorithm, it is compared to the standard C-EGO algorithm proposed by Schonlau (1997) as well as the PC-EGO algorithm by Parr *et al.* (2012). The standard C-EGO algorithm and the two PC-EGO algorithms are competed on the 16 test problems. For the two PC-EGO algorithms, the number of updating points selected in each cycle q is set to 5 and 10 in order to study the influence of q values.

Table 1. Information concerning the selected test problems.

Function name	d	c	Initial design	Maximum iteration	Maximum evaluation $q=1$	$q=5$	$q=10$
Branin	2	1	20	20	40	120	220
G08	2	2	20	20	40	120	220
G24	2	2	20	20	40	120	220
TMF	3	2	30	30	60	180	330
GTCD	4	1	40	40	80	240	440
PVD	4	3	40	40	80	240	440
WBD	4	6	40	40	80	240	440
G13MOD	5	3	50	50	100	300	550
G16	5	38	50	50	100	300	550
Hesse	6	6	60	60	120	360	660
G09	7	4	70	70	140	420	770
SRD	7	11	70	70	140	420	770
G02	10	2	100	100	200	600	1100
G07	10	8	100	100	200	600	1100
G01	13	9	130	100	230	630	1130
G19	15	5	150	100	250	650	1150

The number of initial design points is set to $10d$ for all problems and the initial design points are generated by using the Latin Hypercube Sampling (LHS) method. All three algorithms are allowed to iterate $\min[10d, 100]$ iterations. The maximum number of iterations and the total number of evaluations consumed by each algorithm are listed in Table 1.

The Design and Analysis of Computer Experiment (DACE) toolbox is used to build the Kriging models for both the objective and constraint functions with regression function *regpoly0* and correlation function *corrgauss*. The initial guess of the θ_i $(i = 1, 2, \ldots, d)$ is set to one and the region is set to $[10^{-3}, 10^3]$.

Both the standard CEI criterion and the proposed PCEI criterion are optimized using the Particle Swarm Optimization (PSO) algorithm from MATLAB® 2016b global optimization toolbox with a swarm size of 100 and 100 generations. For the bi-objective criterion of Parr *et al.* (2012), the Non-dominated Sorting Genetic Algorithm II (NSGA-II) algorithm (Deb *et al.* 2002) is used to identify the Pareto optimal solutions. The population size of NSGA-II is set to 100 and the maximum generation is set to 100. After the bi-objective optimization, the obtained Pareto optimal solutions are clustered into q clusters in the design space. Then the point with highest CEI value in each cluster is picked to be the updating point according to the method of Parr *et al.* (2012).

All the experiments are run 30 times with 30 different initial design points to give more reliable results.

4.3. Comparison with EGO algorithms

Figure 2 plots the iteration histories of the three compared algorithms on the 16 test problems. The horizontal axis represents the iteration number and the vertical axis represents the best feasible objective value (average of 30 runs). The plots are started when all the 30 runs have found at least one feasible solution. As a result, different algorithms may start at different iteration numbers in these figures.

First, compared to the standard C-EGO algorithm (dotted line with circles), the proposed PC-EGO algorithm (solid line with squares and diamonds) converges significantly more quickly on all the test problems. The proposed PC-EGO algorithm also gets more optimal results at the end of iteration on most of the test problems. This implies that the proposed PCEI criterion is more efficient than the standard CEI criterion by selecting and evaluating multiple updating points in one iteration. As a result, the proposed PC-EGO algorithm should be preferred when parallel computing architecture is available. As the number of updating points q increases, the proposed PC-EGO algorithm becomes more efficient. It can be seen that the 10-point PCEI criterion converges more quickly than the 5-point criterion and finds better solutions on most of the test problems. This implies that adding more points in each iteration can enhance the searching ability of the PC-EGO algorithm.

Then, compared to the bi-objective CEI criterion (dashed line with squares and diamonds) of Parr *et al.* (2012), it can be found that the proposed PCEI criterion performs significantly better on most of the test problems. The PCEI criterion often converges more quickly and finds more optimal solutions in the end than the bi-objective CEI criterion. The proposed PCEI criterion outperforms the bi-objective CEI criterion significantly on the G08, TMF, WBD, G13MOD, G16, Hesse, G02 and G01 problems. On the G08, TMF, WBD, G16 and Hesse test problems, the bi-objective CEI criterion performs even worse than the standard CEI criterion, while the proposed PCEI criterion performs better than the standard CEI on these test problems.

The bi-objective criterion selects candidate points from a set of Pareto solutions. However, the number of Pareto solutions found by a bi-objective optimizer is uncertain and changes during the iteration process. When the number of obtained Pareto solutions is less than the required number q, additional updating points must be added by using another infill criterion such as the maximizing Kriging standard error function. When the number of obtained Pareto solutions is larger than the required number q, which is often the case, the candidate points have to be carefully selected from the Pareto solutions. Parr *et al.* (2012) suggested clustering the solutions into q clusters in the design

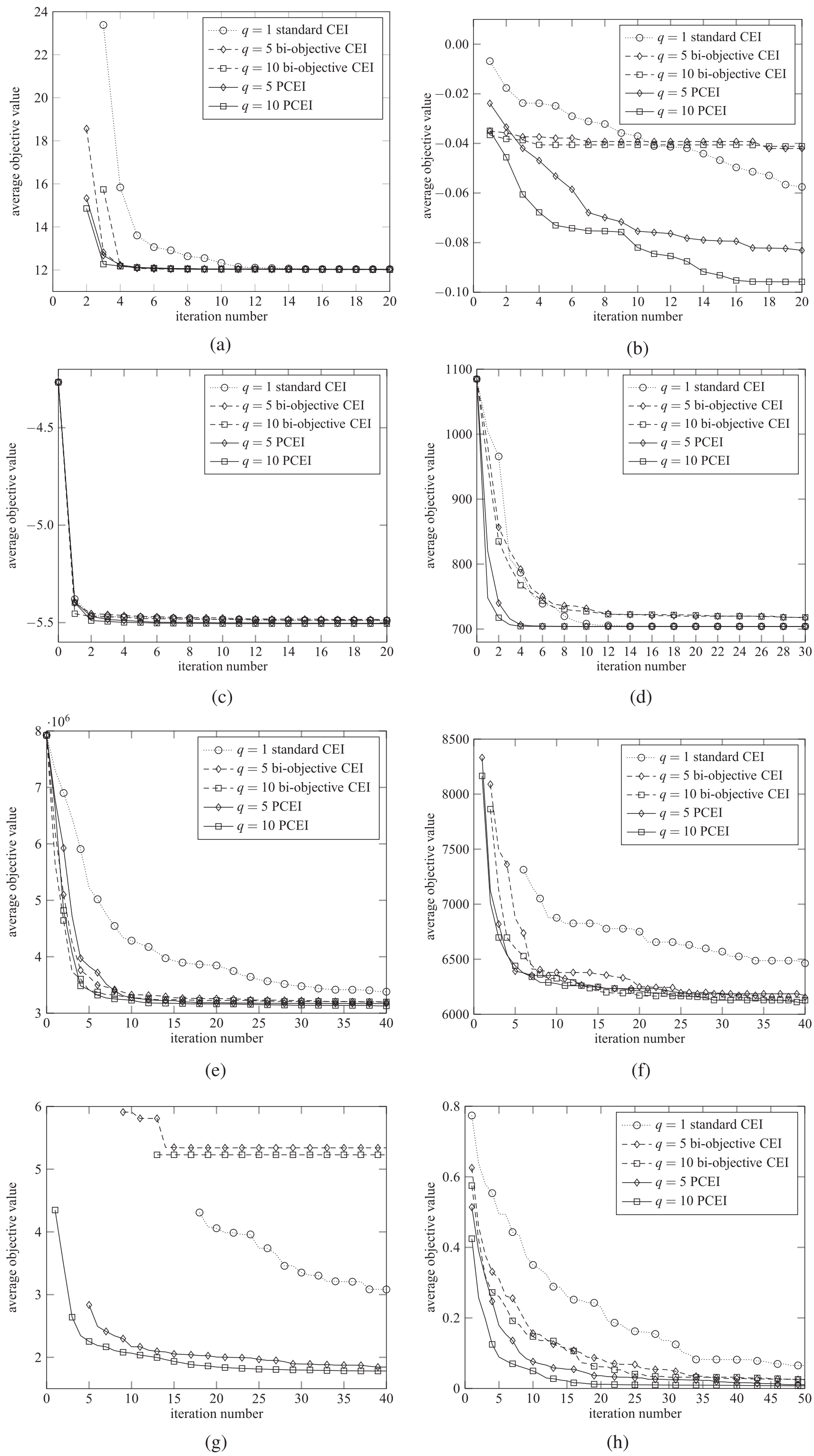


Figure 2. The iteration history on 16 test problems. (a) Branin function. (b) G08 function. (c) G24 function. (d) TMF problem. (e) GTCD problem. (f) PVD problem. (g) WBD problem. (h) G13MOD function. (i) G16 function. (j) Hesse problem. (k) G09 function. (l) SRD problem. (m) G02 function. (n) G07 function. (o) G01 function and (p) G19 function.

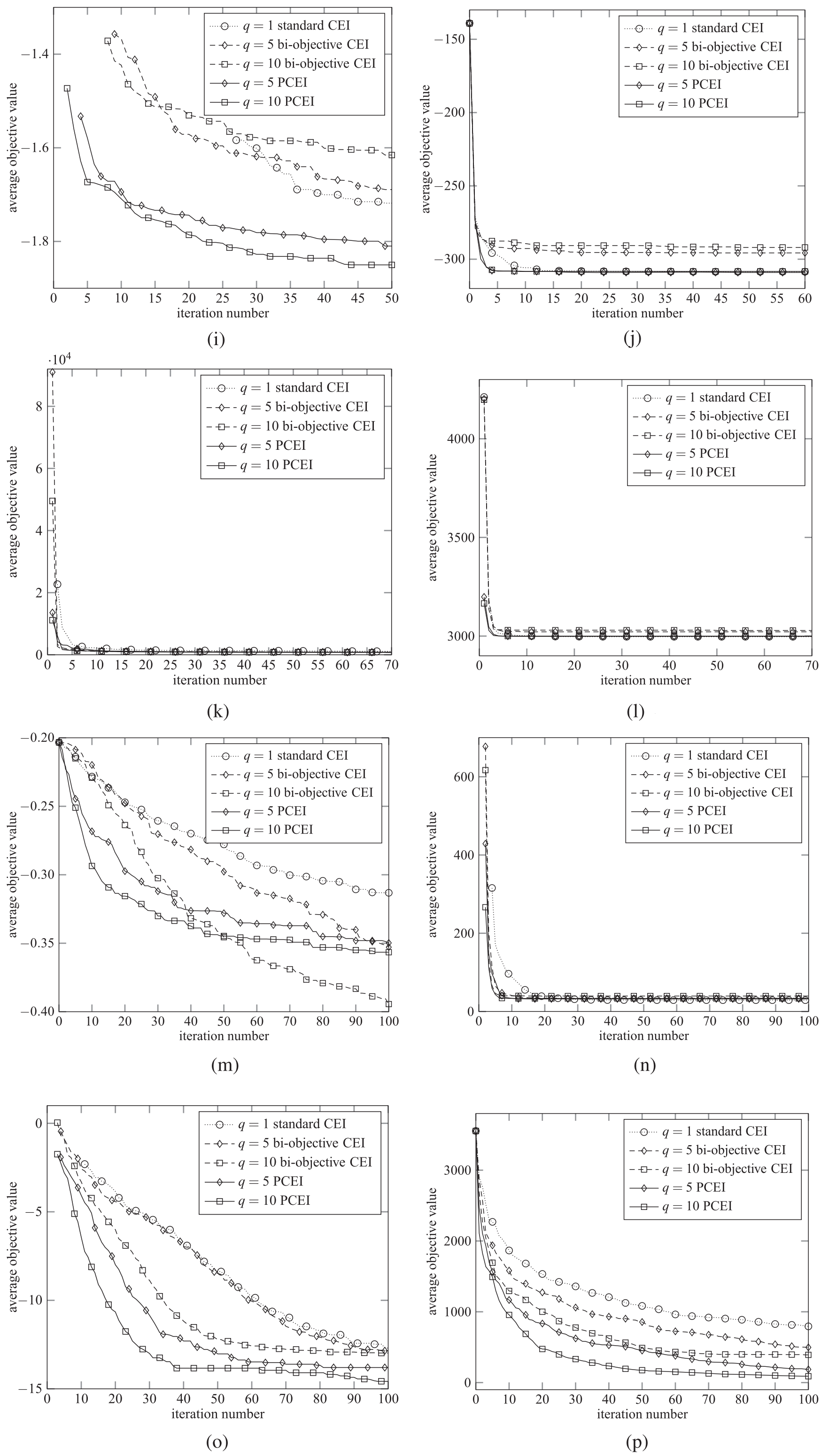


Figure 2. Continued.

Table 2. The final results obtained by the compared EGO algorithms.

		$q = 1$	$q = 5$		$q = 10$	
Function name	Known optimum	CEI	Bi-objective CEI	PCEI	Bi-objective CEI	PCEI
Branin	12.001	12.040 (0.035)	12.024 (0.018)	12.034 (0.027)	12.016 (0.016)	12.022 (0.015)
G08	−0.0958	−0.058 (0.026)	−0.042 (0.028)	**−0.083 (0.021)**	−0.041 (0.023)	**−0.096 (0.000)**
G24	−5.5080	−5.490 (0.022)	−5.484(0.019)	**−5.504 (0.006)**	−5.488 (0.016)	**−5.506 (0.003)**
TMF	703.916	704.10 (0.21)	717.28 (8.01)	**704.00 (0.01)**	718.09 (9.75)	**703.98 (0.04)**
GTCD	2.96E6	3.38E6 (0.39E6)	3.20E6 (0.11E6)	3.17E6(0.21E6)	3.20E6 (0.11E6)	**3.13E6 (0.05E6)**
PVD	5804.45	6463.4 (469.3)	6154.6 (235.1)	6170.6 (221.5)	6127.0 (163.3)	6099.2 (155.5)
WBD	1.725	3.083 (1.245)	5.341 (2.187)	**1.845 (0.069)**	5.229 (2.470)	**1.782 (0.031)**
G13MOD	0.0035	0.065 (0.041)	0.024 (0.022)	**0.011 (0.008)**	0.026 (0.018)	**0.008 (0.004)**
G16	−1.9052	−1.718 (0.191)	−1.689 (0.075)	**−1.810 (0.085)**	−1.616 (0.140)	**−1.850 (0.061)**
Hesse	−310	−308.54 (2.02)	−295.71 (7.89)	**−309.08 (1.25)**	−292.00 (9.78)	**−308.33 (2.16)**
G09	680.6301	1077.7 (166.0)	831.2 (96.1)	817.9 (72.8)	756.4 (30.7)	**740.7 (25.4)**
SRD	2994.42	2996.6 (1.6)	3021.6 (12.6)	**2998.5(3.7)**	3028.6 (15.7)	**3000.0 (3.36)**
G02	−0.4	−0.313 (0.056)	−0.353 (0.077)	−0.350 (0.079)	−0.394 (0.076)	−0.357 (0.086)
G07	24.3062	29.000 (2.272)	33.837 (3.799)	32.159 (4.311)	39.522 (5.906)	**33.295 (4.272)**
G01	−15	−12.708 (1.239)	−12.849 (0.452)	**−13.796 (0.793)**	−12.976 (0.428)	**−14.579 (0.362)**
G19	32.6556	794.19(228.21)	497.39 (132.07)	**185.89 (53.49)**	391.70 (144.80)	**89.91 (14.67)**

Note: The results in boldface are the significantly better (smaller) results between the bi-objective CEI and the PCEI.

space and selecting the point with the highest CEI value from each cluster. The candidates may be distributed equally in the design space, but they may still cluster in the objective space (the EI versus PoF space). On the contrary, the number of candidate points the proposed PCEI criterion produces is always certain. Neither adding supplementary points by using other infill criterion nor selecting from a large number of candidates is needed when using the proposed PCEI criterion. Additionally, the updating point selection problem of the bi-objective CEI criterion is a multiobjective optimization problem and is a single-objective optimization problem for the proposed PCEI criterion. In most cases, the single-objective optimization problem is easier to solve.

In order to give more scientific conclusions, the final results obtained by the standard CEI criterion, the bi-objective CEI criterion and the proposed PCEI criterion are given in Table 2. The best feasible solutions found by each algorithm are displayed. The number outside the blankets is the average value of 30 runs and the number inside the blankets is the standard derivation. The paired t-test between the results of bi-objective CEI criterion and the proposed PCEI criterion is run with respect to $q = 5$ and $q = 10$. When the p value of the significance test is smaller than the significance level $\alpha = 0.05$, it means there is significant difference between the results. The significant better (smaller) results are highlighted with grey background colour. The results in Table 2 show that the proposed PCEI criterion gets significantly better solutions than the bi-objective CEI criterion on 10 out of 16 test problems when $q = 5$, and on 13 out of 16 test problems when $q = 10$. On the rest test problems, the PCEI criterion finds competitive solutions compared to the bi-objective CEI criterion.

4.4. Running time

The running times of the C-EGO by Schonlau (1997), the PC-EGO by Parr *et al.* (2012) and the PC-EGO proposed in this article are compared on the 10*d* G02 problem. The computation time required by each algorithm is shown in Figure 3 with respect to the number of iterations.

It can be seen that the two PC-EGO algorithms need more computation time than the standard C-EGO algorithm, and the 10-point PC-EGOs need more computation time than the 5-point PC-EGOs. In each iteration, the running time of each EGO algorithm is mainly consumed for building Kriging models and optimizing the infill criterion. Compared to the standard C-EGO algorithm, the PC-EGO algorithms collect more samples in each iteration, thus need more time to build the Kriging models.

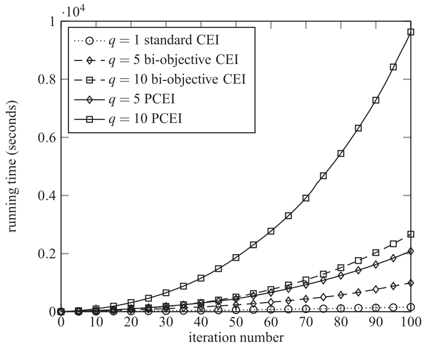


Figure 3. The running time of the compared EGO algorithms.

Compared to the PC-EGO of Parr *et al.* (2012), the proposed PC-EGO algorithm spends more time to select q candidate points. The reason is that the PC-EGO of Parr *et al.* (2012) needs only one multi-objective optimization to obtain q points while the proposed PC-EGO has to run q sequential optimizations. The total time of the proposed 10-point PC-EGO to run 100 iterations is 9629 seconds (around 2.67 hours) on the $10d$ problem, which is significantly less than the time to do 1000 expensive simulations (days or months). Therefore, the computation time for building Kriging models and selecting candidate points is acceptable when applying the proposed PC-EGO algorithm to real-world applications.

4.5. Comparison with the a RBF algorithm

The Filter-based Sequential Radial Basis Function (FSRBF) (Shi *et al.* 2018) is a state-of-the-art RBF algorithm with an advanced constraint handling technique. The FSRBF algorithm has the ability to produce multiple candidate points for parallel evaluation, thus is very suitable for benchmarking the proposed PC-EGO algorithm. To get a fair comparison, the experiment settings are taken from Shi *et al.* (2018). In this experiment, the number of initial design points and the number of updating points used in each iteration q are changed according the dimensions of the test problems. The initial design points are generated using the LHS method, and all the experiments are run ten times using ten different initial designs. The total number of function evaluations is set to 100.

The final results obtained by the proposed PC-EGO and the FSRBF are given in Table 3, where the results of FSRBF are directly cited from Shi *et al.* (2018). It can be seen that the proposed PC-EGO algorithm is comparable to the FSRBF algorithm on the selected test problems. The PC-EGO

Table 3. The experimental results obtained by the FSRBF and the proposed PC-EGO algorithms.

		FSRBF			PC-EGO		
Function name	Global optimum	Best	Median	Mean	Best	Median	Mean
G06	−6961.8139	−6963.03	−6962.08	−6961.86	−6689.94	−6103.82	−6037.43
G08	−0.0958	−0.10	−0.10	−0.09	−0.09	−0.06	−0.05
PVD	5804.45	6149.86	6677.22	6791.01	5992.30	6314.50	6303.50
G05	5126.50	5126.10	5126.49	5126.44	5152.24	5279.50	5323.25
Hesse	−310.00	−310.00	−289.39	−287.60	−309.99	−309.61	−308.93
G04	−30,665.639	−30,522.77	−30,491.47	−30,491.56	−30,663.93	−30,660.29	−30,656.05
G09	680.6301	758.18	1599.32	1476.32	889.75	1155.80	1237.10
SRD	2994.42	2990.54	2997.18	2997.87	2994.70	2997.80	2998.50
G02	−0.40	−0.40	−0.25	−0.27	−0.36	−0.27	−0.28
G07	24.3062	24.32	32.30	42.01	29.77	38.47	54.54
G03MOD	−0.69	−0.01	−0.00	−0.00	−0.00	0.00	−0.00

Table 4. The number of iterations and evluations that the compared algorithms need to find at least one feasible solution.

				C-EGO	PC-EGO	
Metric	Function name	COBRA	Extended ConstrLMSRBF	$q = 1$	$q = 5$	$q = 10$
Number of iterations	G08	3.47 (0.15)	2.20 (0.18)	3.43 (0.68)	2.17 (0.38)	2.13 (0.35)
	PVD	2.87 (0.35)	5.40 (0.65)	11.63 (9.93)	3.07 (1.62)	2.00 (0.79)
	WBD	32.40 (5.92)	20.00 (4.10)	4.90 (3.19)	2.20 (1.06)	1.80 (0.55)
	G13MOD	3.37 (0.78)	2.60 (0.67)	4.60 (3.27)	2.33 (0.92)	2.10 (0.66)
	G16	8.70 (2.37)	13.57 (1.78)	13.30 (10.06)	9.17 (12.85)	5.30 (4.48)
	G09	13.50 (1.85)	15.10 (2.27)	7.73 (4.10)	3.27 (1.14)	2.50 (0.86)
	SRD	1.74 (0.11)	4.40 (0.36)	2.20 (0.96)	1.67 (0.88)	1.40 (0.56)
	G07	36.47 (4.64)	28.83 (2.90)	10.43 (3.79)	6.10 (0.95)	4.43 (1.04)
	G01	1.00 (0.00)	5.07 (0.38)	5.80 (2.87)	3.67 (1.69)	2.92 (1.60)
Number of evaluations	G08	6.47 (0.15)	5.20 (0.18)	6.43 (0.68)	13.83 (1.90)	24.33 (3.46)
	PVD	7.87 (0.35)	10.40 (0.65)	16.63 (0.68)	20.33 (8.09)	25.00 (7.88)
	WBD	37.40 (5.92)	25.00 (4.10)	9.90 (3.19)	16.00 (5.32)	23.00 (5.51)
	G13MOD	9.37 (0.78)	8.60 (0.67)	10.60 (3.27)	17.67 (4.61)	27.00 (6.62)
	G16	14.70 (2.37)	19.57 (1.78)	19.30 (10.06)	51.83 (64.25)	59.00 (44.81)
	G09	21.50 (1.85)	23.10 (2.27)	15.73 (4.10)	24.33 (5.71)	33.00 (8.61)
	SRD	9.47 (0.11)	12.40 (0.36)	10.20 (0.96)	16.33 (4.42)	22.00 (5.63)
	G07	47.47 (4.64)	39.83 (2.90)	21.43 (3.79)	41.50 (9.77)	55.33 (10.40)
	G01	15.00 (0.00)	19.07 (0.38)	19.80 (2.87)	32.33 (8.44)	42.33 (15.99)

algorithm outperforms the FSRBF on the PVD, Hesse, G04 and G09 problems in terms of the mean feasible objective. On the G06, G05 and SRD problems, the best result obtained by FSRBF is even smaller than the best known optimum because the FSRBF used a margin-to-accept solution with slight constraint violation. In comparison, the proposed PC-EGO algorithm did not use a constraint margin and is still able to find near-optimum solutions on these problems. The proposed PC-EGO algorithm performs slightly worse on the G08, G05 and G07 problems than the FSRBF algorithm. Overall, the proposed PC-EGO algorithm is competitive with the FSRBF algorithm when only 100 function evaluations are allowed.

4.6. Comparison when all initial points are infeasible

In order to study the ability of the proposed PC-EGO algorithm to find the feasible regions, additional experiments are conducted using all infeasible initial design points. The experiment settings are the same as in Section 4.2 except that the number of initial samples is $n + 1$. The number of iterations and the number of function evaluations are counted once a feasible solution is found. The results are shown in Table 4, where the results of COBRA and Extended ConstrLMSRBF are cited from Regis (2014).

It can be seen in Table 4 that the C-EGO and PC-EGO algorithms are able to find the feasible regions on all test problems. The C-EGO algorithm shows competitive ability to find the feasible regions compared to the COBRA and Extended ConstrLMSRBF algorithms (Regis 2014). Compared to the C-EGO algorithm, the PC-EGO algorithm is able to accelerate the process by selecting and evaluating multiple samples in each iteration. As the table shows, the number of iterations that the PC-EGO needs to find a feasible solution decreases gradually as the number of selected samples q increases. However, the total number of function evaluations that the PC-EGO consumes is greater than the C-EGO algorithm. Therefore, the PC-EGO algorithm has the advantage of finding feasible solutions when parallel computing resources are available.

5. Optimal design of a ring-stiffened conical shell

Finally, the proposed PC-EGO algorithm is applied to the design of a ring-stiffened conical shell. The structure of the stiffened shell is shown in Figure 4. The shell thickness and stiffener size in the first

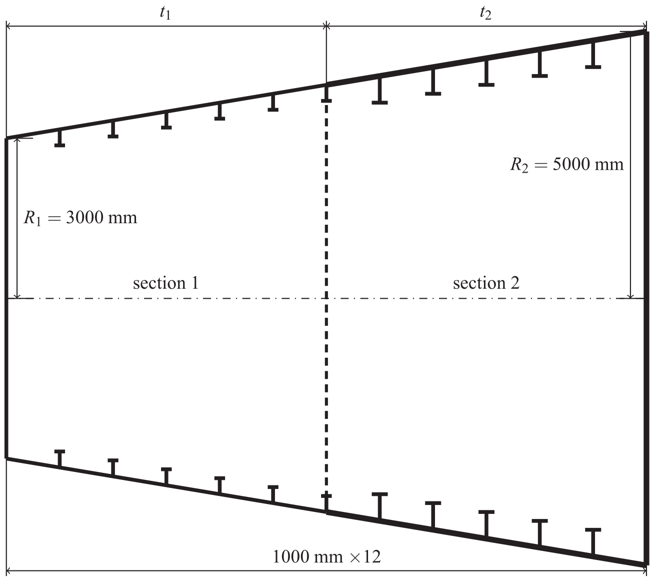


Figure 4. The stiffened conical shell.

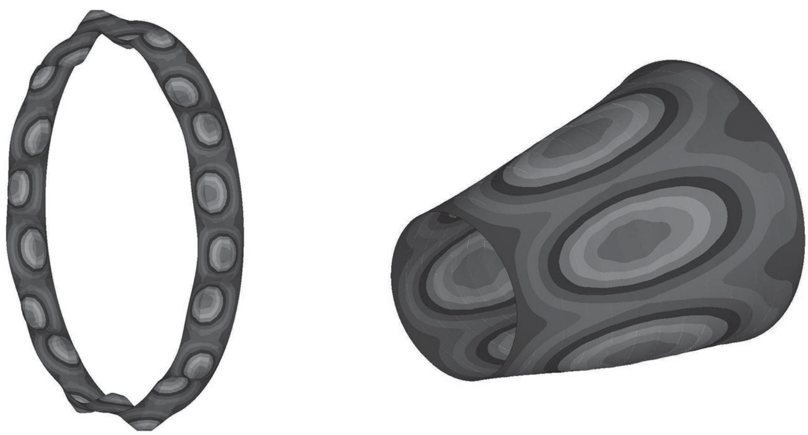
six frame spaces (section 1) are the same, and they are also the same in the second six frame spaces (section 2). But the thickness and stiffener size are different between the first six frame spaces and the second six frame spaces. The elastic modulus of the material is $E = 2.1 \times 10^{11}$ Pa, the Poisson ratio is $p = 0.3$, the density is $\rho = 7850\,\mathrm{kg\,m^{-3}}$ and the yield limit is $\sigma_S = 650$ MPa. The calculation pressure is $p_j = 6.6$ MPa. The design variables of the stiffened conical shell are the thicknesses of the shells and the sizes of the stiffeners. The goal is to maximize the critical global buckling pressure of the shell. The constraints include three stress constraints, two critical buckling pressure constraints and one weight constraint. The optimization problem of the conical shell can be described as follows.

$$\begin{aligned}
\text{Find} \quad & \mathbf{x} = [x_1, x_2, \ldots, x_{10}] \\
\text{maximize} \quad & f = p_{\mathrm{cr1}} \\
\text{subjectto} \quad & g_1 = \frac{\sigma_1}{0.85\sigma_S} - 1 \le 0 \\
& g_2 = \frac{\sigma_2}{1.15\sigma_S} - 1 \le 0 \\
& g_3 = \frac{\sigma_3}{0.6\sigma_S} - 1 \le 0 \\
& g_4 = 1 - \frac{p_{\mathrm{cr1}}}{1.2p_j} \le 0 \\
& g_5 = 1 - \frac{p_{\mathrm{cr2}}}{p_j} \le 0 \\
& g_6 = \frac{w}{160{,}000\,\mathrm{kg}} - 1 \le 0,
\end{aligned} \tag{15}$$

where $x_1, x_2, \ldots, x_{10}$ are design variables, whose meanings and the design space are described in Table 5; p_{cr1} is the critical global (overall) buckling pressure of the stiffened shell; σ_1, σ_2 and σ_3 are the circumferential stress at the midpoint of stiffener, the longitudinal stress at the end of stiffener and the stress in the stiffener, respectively; p_{cr2} is the critical local (shell) buckling pressure of the

Table 5. The design space of the ring-stiffened conical shell optimization problem.

Symbol	Description	Unit	Lower bound	Upper bound
x_1	Height of stiffener web in section 1	mm	200	340
x_2	Thickness of stiffener web in section 1	mm	10	24
x_3	Width of stiffener flange in section 1	mm	100	240
x_4	Thickness of stiffener flange in section 1	mm	10	24
x_5	Height of stiffener web in section 2	mm	200	340
x_6	Thickness of stiffener web in section 2	mm	10	24
x_7	Width of stiffener flange in section 2	mm	100	240
x_8	Height of stiffener web in section 2	mm	10	24
x_9	Shell thickness in section 1	mm	50	64
x_{10}	Shell thickness in section 2	mm	50	64



(a) (b)

Figure 5. The local and global buckling mode shapes of the stiffened shell. (a) The local buckling mode shape and (b) The global buckling mode shape.

stiffened shell; w is the overall weight of the stiffened shell. The critical buckling pressures and the stresses are calculated using the computationally expensive finite element method. A local buckling mode shape and a global buckling mode shape of the stiffened shell are shown in Figure 5. Since both the objective and constraints of the optimization problem involve computationally expensive finite element analysis, this problem is a typical computationally expensive constrained optimization problem.

The standard C-EGO algorithm, PC-EGO algorithm of Parr *et al.* (2012) as well as the proposed PC-EGO algorithm are run to solve the ring-stiffened conical shell optimization problem. The Latin hypercube sampling method is used to generate 100 initial designs. Then the 100 initial designs are evaluated using the finite element method to obtain their objective and constraint values. After that, the standard C-EGO and the two PC-EGO algorithms are executed to search for the global feasible solution of the problem. The settings of the internal optimization are the same as the settings in Section 4.2. The standard C-EGO and the two PC-EGO algorithms are allowed to process 100 iterations.

The iteration histories of the standard C-EGO algorithm and the two PC-EGO algorithms are shown in Figure 6. It is clear that the two parallel CEI criteria have faster convergence speed than the standard CEI criterion, and that the proposed PCEI criterion converges faster than the bi-objective PCEI criterion on the ring-stiffened conical shell optimization problem. At the end of 100 iterations, the two PC-EGO algorithms find much higher critical global buckling pressure values than the standard C-EGO algorithm, and the proposed PC-EGO algorithm finds better solutions than the PC-EGO

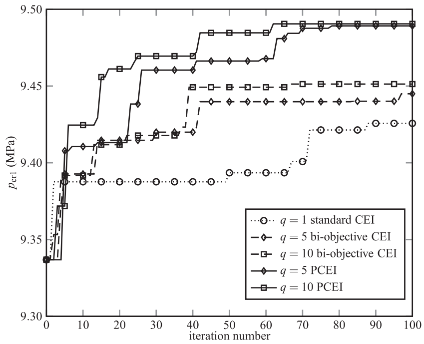


Figure 6. The iteration history of the ring-stiffened conical shell problem.

algorithm of Parr *et al.* (2012). It is interesting to see that the proposed 10-point PC-EGO algorithm converges faster than the 5-point PC-EGO algorithm, but the final objective values obtained by the 10-point PC-EGO algorithm are only slightly better than the 5-point PC-EGO algorithm on this ring-stiffened conical shell optimization problem. This means that using more parallel computing resources have greater advantages if the optimization process is stopped at earlier iterations. When the iterations go on for long enough, both the 5-point and the 10-point PC-EGO algorithms can converge to the global optimum on the ring-stiffened conical shell optimization problem.

The final optimization results obtained by each algorithm are given in Table 6. The design variables, the final objective value and the constraint values of the the ring-stiffened conical shell optimization problem are displayed. It can be seen that the constraints σ_2, σ_3 and w are very close to their boundaries at the optimal solutions. This means that these three constraints are active constraints of the optimization problems while the other constraints are not. The proposed PC-EGO algorithm finds higher critical global buckling pressures than the PC-EGO algorithm of Parr *et al.* (2012) at the end of 100 iterations for both $q = 5$ and $q = 10$. In conclusion, the proposed PC-EGO algorithm can speed up the search significantly and find better solutions at the end of the search process compared to the standard C-EGO algorithm on the ring-stiffened conical shell optimization problem. Additionally, the proposed algorithm also show faster convergence speed and competitive search ability when compared to the PC-EGO algorithm of Parr *et al.* (2012).

Table 6. The optimization results obtained by the compared algorithms on the ring-stiffened conical shell problem.

				$q = 1$	$q = 5$		$q = 10$	
Variable type	Symbol	Unit	Range	CEI	Bi-objective CEI	PCEI	Bi-objective CEI	PCEI
Design variables	x_1	mm	[200, 340]	204.03	213.75	340.00	258.49	339.98
	x_2	mm	[10, 24]	22.364	19.500	24.000	23.454	23.997
	x_3	mm	[100, 240]	229.02	227.11	240.00	142.39	240.00
	x_4	mm	[10, 24]	23.047	20.341	24.000	10.099	24.000
	x_5	mm	[200, 340]	339.83	245.47	239.60	327.97	203.39
	x_6	mm	[10, 24]	24.000	23.968	23.943	23.944	24.000
	x_7	mm	[100, 240]	100.05	224.81	100.00	158.05	100.00
	x_8	mm	[10, 24]	10.004	23.031	12.214	19.474	23.934
	x_9	mm	[50, 64]	54.185	54.804	50.403	57.105	50.339
	x_{10}	mm	[50, 64]	61.992	60.980	64.000	60.415	63.765
Objective	p_{cr1}	MPa	≥ 7.92	9.4257	9.4449	9.4892	9.4513	9.4905
Constraints	σ_1	MPa	≤ 552.5	454.15	451.87	451.47	456.25	451.44
	σ_2	MPa	≤ 747.5	732.65	741.49	714.62	747.36	716.59
	σ_3	MPa	≤ 390	388.58	389.40	389.97	389.90	389.42
	p_{cr1}	MPa	≥ 7.92	9.4257	9.4449	9.4892	9.4513	9.4905
	p_{cr2}	MPa	≥ 6.6	9.2313	8.9488	8.0278	8.7949	7.9999
	w	kg	$\leq 160{,}000$	159,599	159,949	159,996	159,983	159,985

6. Conclusions

This article proposed a new Parallel Constrained Efficient Global Optimization (PC-EGO) algorithm for optimization problems with computationally expensive objective and constraints. The proposed PC-EGO algorithm uses the proposed Pseudo Constrained Expected Improvement (PCEI) criterion to select multiple updating points for parallel computing within one iteration. The proposed PC-EGO algorithm is tested on 16 analytical problems against the standard C-EGO algorithm as well as another state-of-the-art PC-EGO algorithm. The results indicate that the proposed PC-EGO algorithm is able to speed up the standard C-EGO algorithm significantly. When compared to the state-of-the-art PC-EGO algorithm, the proposed PC-EGO algorithm performs more efficiently and robustly on the test problems. Moreover, when the proposed PC-EGO algorithm is applied to the optimal design of a ring-stiffened conical shell, it converges significantly faster and gets significantly better solutions than the standard C-EGO algorithm. The proposed algorithm also shows competitive performance when compared to the state-of-the-art PC-EGO algorithm. This implies that the proposed PC-EGO algorithm has more potential for solving computationally expensive constrained optimization problems when parallel computing architecture is available.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Dawei Zhan http://orcid.org/0000-0002-0173-3447

References

Arora, J. S1989. *Introduction to Optimum Design*. New York: McGraw-Hill Higher Education.

Audet, C., J. Denni, D. Moore, A. Booker, and P. Frank. 2000. “A Surrogate-Model-Based Method for Constrained Optimization.” In *Proceedings of the 8th Symposium on Multidisciplinary Analysis and Optimization*. Reston, VA: AIAA. doi:10.2514/6.2000-4891.

Beightler, C. S., and D. T. Phillips. 1976. *Applied Geometric Programming*. New York: Wiley.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II.” *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197.

Duan, X., G. G. Wang, X. Kang, Q. Niu, G. Naterer, and Q. Peng. 2009. “Performance Study of Mode-Pursuing Sampling Method.” *Engineering Optimization* 41 (1): 1–21.

Feng, Z. W., Q. B. Zhang, Q. F. Zhang, Q. G. Tang, T. Yang, and Y. Ma. 2015. “A Multiobjective Optimization Based Framework to Balance the Global Exploration and Local Exploitation in Expensive Optimization.” *Journal of Global Optimization* 61 (4): 677–694.

Floudas, C. A., and P. M. Pardalos. 1990. *A Collection of Test Problems for Constrained Global Optimization Algorithms*. Berlin: Springer-Verlag.

Forrester, A. I. J., and A. J. Keane. 2009. “Recent Advances in Surrogate-Based Optimization.” *Progress in Aerospace Sciences* 45 (1): 50–79.

Forrester, A. I. J., A. Sóbester, and A. J. Keane. 2008. *Engineering Design via Surrogate Modelling: A Practical Guide*. Chichester, UK: Wiley.

Ginsbourger, D., R. L. Riche, and L. Carraro. 2010. “Kriging Is Well-Suited to Parallelize Optimization.” In *Computational Intelligence in Expensive Optimization Problems*, edited by Yoel Tenne and Chi-Keong Goh, 131–162. Berlin: Springer.

Gramacy, R. B., G. A. Gray, S. L. Digabel, H. K. H. Lee, P. Ranjan, G. Wells, and S. M. Wild. 2016. “Modeling An Augmented Lagrangian for Blackbox Constrained Optimization.” *Technometrics* 58 (1): 1–11.

Hesse, R. 1973. “A Heuristic Search Procedure for Estimating A Global Solution of Nonconvex Programming Problems.” *Operations Research* 21: 1267–1280.

Jie, H., Y. Wu, and J. Ding. 2015. “An Adaptive Metamodel-Based Global Optimization Algorithm for Black-Box Type Problems.” *Engineering Optimization* 47 (11): 1459–1480.

Jin, Y. 2011. “Surrogate-Assisted Evolutionary Computation: Recent Advances and Future Challenges.” *Swarm and Evolutionary Computation* 1 (2): 61–70.

Jin, Y., M. Olhofer, and B. Sendhoff. 2002. “A Framework for Evolutionary Optimization with Approximate Fitness Functions.” *IEEE Transactions on Evolutionary Computation* 6 (5): 481–494.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient Global Optimization of Expensive Black-Box Functions." *Journal of Global Optimization* 13 (4): 455–492.
Kazemi, M., G. G. Wang, S. Rahnamayan, and K. Gupta. 2011. "Metamodel-Based Optimization for Problems with Expensive Objective and Constraint Functions." *Journal of Mechanical Design* 133: 014505-1–014505-7.
Liang, J. J., T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. Coello Coello, and K. Deb. 2006. "Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization." Technical Report, Nanyang Technological University, Singapore.
Liu, H., S. Xu, X. Chen, X. Wang, and Q. Ma. 2017. "Constrained Global Optimization Via a DIRECT-Type Constraint-Handling Technique and An Adaptive Metamodeling Strategy." *Structural and Multidisciplinary Optimization* 55 (1): 155–177.
Long, T., D. Wu, X. S. Guo, G. G. Wang, and L. Liu. 2015. "Efficient Adaptive Response Surface Method Using Intelligent Space Exploration Strategy." *Structural and Multidisciplinary Optimization* 51 (6): 1335–1362.
Michalewicz, Z., and M. Schoenauer. 1996. "Evolutionary Algorithms for Constrained Parameter Optimization Problems." *Evolutionary Computation* 4 (1): 1–32.
Müller, J., and C. A. Shoemaker. 2014. "Influence of Ensemble Surrogate Models and Sampling Strategy on the Solution Quality of Algorithms for Computationally Expensive Black-Box Global Optimization Problems." *Journal of Global Optimization* 60 (2): 123–144.
Müller, J., and J. D. Woodbury. 2017. "GOSAC: Global Optimization with Surrogate Approximation of Constraints." *Journal of Global Optimization* 69 (1): 117–136.
Parr, J. M., A. J. Keane, A. I. J. Forrester, and C. M. E. Holden. 2012. "Infill Sampling Criteria for Surrogate-Based Optimization with Constraint Handling." *Engineering Optimization* 44 (10): 1147–1166.
Rao, S. S. 1996. *Engineering Optimization*. 3rd ed. New York: Wiley.
Rashid, K., S. Ambani, and E. Cetinkaya. 2013. "An Adaptive Multiquadric Radial Basis Function Method for Expensive Black-Box Mixed-Integer Nonlinear Constrained Optimization." *Engineering Optimization* 45 (2): 185–206.
Regis, Rommel G. 2011. "Stochastic Radial Basis Function Algorithms for Large-Scale Optimization Involving Expensive Black-Box Objective and Constraint Functions." *Computers & Operations Research* 38 (5): 837–853. doi:10.1016/j.cor.2010.09.013.
Regis, R. G. 2014. "Constrained Optimization by Radial Basis Function Interpolation for High-Dimensional Expensive Black-Box Problems with Infeasible Initial Points." *Engineering Optimization* 46 (2): 218–243. doi:10.1080/0305215X.2013.765000.
Regis, R. G., and C. A. Shoemaker. 2005. "Constrained Global Optimization of Expensive Black Box Functions Using Radial Basis Functions." *Journal of Global Optimization* 31 (1): 153–171.
Regis, R. G., and C. A. Shoemaker. 2013. "Combining Radial Basis Function Surrogates and Dynamic Coordinate Search in High-Dimensional Expensive Black-Box Optimization." *Engineering Optimization* 45 (5): 529–555.
Saad, A., Z. Dong, B. Buckham, C. Crawford, A. Younis, and M. Karimi. 2019. "A New Kriging–Bat Algorithm for Solving Computationally Expensive Black-Box Global Optimization Problems." *Engineering Optimization* 51 (2): 265–285. doi:10.1080/0305215X.2018.1461853.
Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn. 1989. "Design and Analysis of Computer Experiments." *Statistical Science* 4 (4): 409–423.
Sasena, M. J., P. Papalambros, and P. Goovaerts. 2002. "Exploration of Metamodeling Sampling Criteria for Constrained Global Optimization." *Engineering Optimization* 34 (3): 263–278.
Schonlau, M. 1997. "Computer Experiments and Global Optimization." PhD thesis, University of Waterloo, Canada.
Shahrokhi, A., and A. Jahangirian. 2010. "A Surrogate Assisted Evolutionary Optimization Method with Application to the Transonic Airfoil Design." *Engineering Optimization* 42 (6): 497–515.
Shi, R., L. Liu, T. Long, Y. Wu, and Y. Tang. 2018. "Filter-Based Sequential Radial Basis Function Method for Spacecraft Multidisciplinary Design Optimization." *AIAA Journal* 57 (3): 1019–1031.
Simpson, T. W., A. J. Booker, D. Ghosh, A. A. Giunta, P. N. Koch, and R.-J. Yang. 2004. "Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion." *Structural and Multidisciplinary Optimization* 27 (5): 302–313.
Sóbester, A., S. J. Leary, and A. J. Keane. 2004. "A Parallel Updating Scheme for Approximating and Optimizing High Fidelity Computer Simulations." *Structural and Multidisciplinary Optimization* 27 (5): 371–383.
Tang, Y., J. Chen, and J. Wei. 2013. "A Surrogate-Based Particle Swarm Optimization Algorithm for Solving Optimization Problems with Expensive Black Box Functions." *Engineering Optimization* 45 (5): 557–576.
Viana, F. A. C., R. T. Haftka, and L. T. Watson. 2013. "Efficient Global Optimization Algorithm Assisted by Multiple Surrogate Techniques." *Journal of Global Optimization* 56: 669–689. doi:10.1007/s10898-012-9892-5.
Viana, F. A. C., T. W. Simpson, V. Balabanov, and V. Toropov. 2014. "Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come?." *AIAA Journal* 52 (4): 670–690.
Wang, G. G. 2003. "Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points." *Journal of Mechanical Design* 125 (2): 210–220.
Wang, G. G., Z. Dong, and P. Aitchison. 2001. "Adaptive Response Surface Method—A Global Optimization Scheme for Approximation-Based Design Problems." *Engineering Optimization* 33 (6): 707–734.

Wang, G. G., and S. Shan. 2007. "Review of Metamodeling Techniques in Support of Engineering Design Optimization." *Journal of Mechanical Design* 129 (2): 370–380.
Wang, L., S. Shan, and G. G. Wang. 2004. "Mode-Pursuing Sampling Method for Global Optimization on Expensive Black-Box Functions." *Engineering Optimization* 36 (4): 419–438.
Younis, A., and Z. Dong. 2010. "Trends, Features, and Tests of Common and Recently Introduced Global Optimization Methods." *Engineering Optimization* 42 (8): 691–718.
Zhan, D., J. Qian, and Y. Cheng. 2017. "Pseudo Expected Improvement Criterion for Parallel EGO Algorithm." *Journal of Global Optimization* 68 (3): 641–662.