



# Tabu efficient global optimization with applications in additive manufacturing

Long Wang<sup>1</sup> · Theodore T. Allen<sup>1</sup> · Michael A. Groeber<sup>1</sup>

Received: 22 May 2020 / Revised: 23 December 2020 / Accepted: 8 January 2021 / Published online: 26 February 2021  
© The Author(s) 2021

## Abstract

Methods based on Gaussian stochastic process (GSP) models and expected improvement (EI) functions have been promising for box-constrained expensive optimization problems. These include robust design problems with environmental variables having set-type constraints. However, the methods that combine GSP and EI sub-optimizations suffer from the following problem, which limits their computational performance. Efficient global optimization (EGO) methods often repeat the same or nearly the same experimental points. We present a novel EGO-type constraint-handling method that maintains a so-called tabu list to avoid past points. Our method includes two types of penalties for the key “infill” optimization, which selects the next test runs. We benchmark our tabu EGO algorithm with five alternative approaches, including DIRECT methods using nine test problems and two engineering examples. The engineering examples are based on additive manufacturing process parameter optimization informed using point-based thermal simulations and robust-type quality constraints. Our test problems span unconstrained, simply constrained, and robust constrained problems. The comparative results imply that tabu EGO offers very promising computational performance for all types of black-box optimization in terms of convergence speed and the quality of the final solution.

**Keywords** Surrogate systems · Kriging · Robust optimization · Constrained optimization · Computer experiments

## 1 Introduction

Engineering design optimization problems often involve expensive black-box functions such as finite element methods (FEM), computational fluid dynamics (CFD), or thermal simulations based on Green’s functions. The computational times required to build and run these simulations could be minutes, hours, or even days. Therefore, methods based on Gaussian stochastic process (GSP) or Kriging meta-models provide computationally inexpensive approaches to speed up the optimization by providing surrogates for more expensive simulations (Matheron 1963; Santner et al. 2018). GSP meta-

models have been incorporated into the so-called efficient global optimization (EGO) to solve unconstrained (Jones et al. 1998), simply constrained (Audet et al. 2000; Sasena et al. 2002; Sasena 2002), and robust constrained (ur Rehman and Langelaar 2017) optimization problems. In all these cases, the EGO algorithm selects the next expensive (in terms of computation time) function evaluation or sample using an “infill” optimization involving the GSP meta-model. These infill optimizations involve some form of expected improvement (EI, Jones et al. 1998) formula or other infill sampling criterion (ISC) such as mode-pursuing sampling (Wang et al. 2004) to balance exploration and exploitation.

EGO applications have been extended to “noisy” problems involving either stochastic simulations or real-world experiments (e.g., Huang and Allen, 2005). The term “noisy” in optimization referring to the problem affected by uncertainties, and it is often called the robust optimization problem. Ben-Tal et al. (2009) summarized the recent progress of robust optimization for linear, convex-quadratic, conic-quadratic, and semidefinite problems. However, literature related to robust optimization of non-convex problems affected by

---

Responsible Editor: Xiaoping Du

✉ Theodore T. Allen  
allen.515@osu.edu

<sup>1</sup> Department of Integrated Systems Engineering, Ohio State University, 210 Baker Systems Building, 1971 Neil Avenue, Columbus, OH 43210, USA

uncertainties is relatively limited. Bertsimas et al. (2010) proposed an algorithm for robust optimization of non-convex problems with constraints, but the method is aimed at identifying local robust optima only. In 2015, Reman and Langelaar proposed the first Kriging-based constrained optimization for addressing efficient global robust optimization of unconstrained problems affected by parametric uncertainties, and they expanded to the constrained optimization problem (ur Rehman and Langelaar 2017). This paper utilizes the framework introduced by Reman and Langelaar to solve the constrained optimization problem with max/min-typed constraints, which will be explained in detail in Section 2.1, by dividing the optimization problem into two sub-optimization problems.

EGO methods often suffer from repeating the same or nearly the same experimental points, which limits their computational performance. Huang et al. (2005) addressed the “diminishing return of addition replicates” in creating a new EI-based infill sampling formula; i.e., they addressed the unfortunate tendency of the EI methods to return to the same points for noisy problems. The associated relatively slow convergence rates were noted even for noisy cases in which repeated sampling could be (hypothetically) beneficial. Those authors created a so-called augmented expected improvement (AEI) to reduce resampling. Here, we propose methods that directly and explicitly increase the probability of avoiding returns to or near previously sampled points using a so-called tabu list. Our proposed lists comprise previously sampled points, and points predicted to be infeasible identified during infill optimizations.

Tabu searches (TS) are highly regarded meta-heuristics for the optimization of inexpensive black-box functions (Glover 1986, 1989, 1990; Chelouah and Patrick 2000). Generating variants of TS methods continues to be an active area of research for inexpensive functions, including new types of “hybrid” optimization (Duarte et al. 2011). TS methods have been combined with other meta-heuristics, such as particle swarm methods (Lin et al. 2019). However, TS methods have not been applied to expensive black-box functions and have not been combined with EGO methods. Although the idea of tabu search could be applied to most of the sequential global optimization methods, combining TS and EGO to address expensive black-box optimization is our primary objective here.

It should be noted that Forrester et al. (2006) also observed the repetition tendency in noisy black-box methods of EGO. They proposed that nontrivial regression models should be included in the Kriging modeling to avoid repetition. Even though our problem here is noiseless, we include the so-called superEGO methods with first-order regression in our comparison.

The method for performing infill optimizations using the inexpensive GSP meta-models is generally not a major

concern in the context of the EGO method because the computational overhead often has less importance than the costs of the expensive black-box runs. Therefore, many types of non-linear programming approaches can be applied, including DIRECT methods (Jones 2009a; Liu et al. 2017), constraint importance mode-pursuing sampling for continuous global optimization (CIMPS, Kazemi et al. 2010), and constrained local metric stochastic radial-based function (ConstrLMSRBF, Regis 2011) methods. In our implementation, we use DIRECT for our infill optimization; a brief introduction of the DIRECT method can be found in Section 2.4. However, non-linear programming approaches have been proposed to apply to expensive black-box functions (e.g., Liu et al. 2017). We also included MATLAB’s build-in constrained non-linear optimization algorithm *fmincon* (The Math Works, Inc 2020), and Tree Parzen Estimator (Bergstra et al. 2013) from a python package *hyperopt* is included in the results comparison study. Therefore, the methods mentioned above are included in our computational comparison together with the pre-existing EGO-type methods.

The motivating application for this work is the optimal design of additive manufacturing process parameters informed by point-based thermal simulations (Frazier 2014; Schwalbach et al. 2019). These thermal simulation outputs derive from a sum over Green’s function evaluations evaluated at discrete points in a physical two-dimensional layer (Schwalbach et al. 2019). This type of simulation differs from usual FEM or CFD simulations in that it is possible to obtain values at discrete  $X$ - $Y$  points only. The FEM and CFD methods almost unavoidably generate outputs at all  $X$ - $Y$  points in the part geometry. Therefore, the use of discrete thermal simulations motivates the application of a robust formulation in which the part thermal quality must be guaranteed at all  $X$ - $Y$  points in the process parameter optimization. This is challenging because potentially the worst  $X$ - $Y$  point might differ depending on the process parameter values. We address the decision problem posed for robust EGO methods (ur Rehman and Langelaar 2017).

The remainder of this article is organized as follows. Section 2 reviews the decision problems in unconstrained (Jones et al. 1998), simply constrained (Audet et al. 2000; Sasena et al. 2002; Sasena 2002), and robust constrained (ur Rehman and Langelaar 2017) optimization and provides a brief introduction to the DIRECT method. The first two problems are described as a special case of the third. Also, the mechanics of GSP meta-models and pseudo-code for EGO methods are reviewed, again with unconstrained and simply constrained methods described as special cases of robust EGO methods. Further, TS method concepts and methods are briefly described. In Section 3, we propose the novel tabu EGO methods

building on robust EGO and concepts from TS methods. Sections 4 and 5 compare the performance of tabu EGO with three alternative methods on nine numerical test problems and two additive manufacturing process parameter optimization problems. Finally, Section 6 discusses the implications and suggests multiple opportunities for future research.

## 2 Review of alternative methods

### 2.1 Decision problems

The general robust decision problem was explored by ur Rehman and Langelaar (2017). The controllable decision variables are  $\mathbf{x}_c$  which are restricted to the set  $\mathbb{X}_c$ . In addition, there are environmental or noise variables,  $\mathbf{x}_e$ , restricted to the set  $\mathbb{X}_e$ . In our additive example, the process parameter variables are controllable, and the noise variables are the  $X$ - $Y$  positions because our part must have acceptable quality at all points in each layer. The objective function is  $f(\mathbf{x}_c, \mathbf{x}_e)$ , and the  $q$  constraints are  $g_j(\mathbf{x}_c, \mathbf{x}_e)$ ,  $j = 1, \dots, q$ . In our example, the objective is the processing speed times, and the constraints related to part quality at all points. The general formulation is

$$\begin{aligned} \min_{\mathbf{x}_c} \max_{\mathbf{x}_e} f(\mathbf{x}_c, \mathbf{x}_e) \\ \text{s.t. } \max_{\mathbf{x}_e} g_j(\mathbf{x}_c, \mathbf{x}_e) \leq 0 \quad \forall j = 1, \dots, q \\ \mathbf{x}_c \in \mathbb{X}_c, \mathbf{x}_e \in \mathbb{X}_e. \end{aligned} \tag{1}$$

Note that “simply” constrained formulations are special cases for which the environmental variable is a constant, i.e.,  $|\mathbb{X}_e| = 1$ . Unconstrained problems are special cases with  $q = 0$ .

Two potentially significant quantities are the worst environmental conditions at iteration  $t$ ,  $\mathbf{x}_{e,t}^*$ , for a given set of control parameters  $\mathbf{x}_c^*$  and the best control parameters at iteration  $t$ ,  $\mathbf{x}_{c,t}^*$ , for a given environmental set of conditions  $\mathbf{x}_e^*$ . These quantities are

$$\mathbf{x}_{e,t}^* = \operatorname{argmax}_{\mathbf{x}_e \in \mathbb{X}_e} f(\mathbf{x}_c^*, \mathbf{x}_e) | g_j(\mathbf{x}_c^*, \mathbf{x}_e) \leq 0 \forall j = 1, \dots, q, \mathbf{x}_e \in \mathbb{X}_e. \tag{2}$$

and

$$\mathbf{x}_{c,t}^* = \operatorname{argmin}_{\mathbf{x}_c \in \mathbb{X}_c} f(\mathbf{x}_c, \mathbf{x}_e^*) | g_j(\mathbf{x}_c, \mathbf{x}_e^*) \leq 0 \forall j = 1, \dots, q, \mathbf{x}_c \in \mathbb{X}_c. \tag{3}$$

The two quantities in (2) and (3) relate to the two sub-optimizations of the robust EGO algorithm (ur Rehman and

Langelaar 2017). For methods without robustness constraints, (2) is not relevant (e.g., Jones et al. 1998; Sasena 2002).

### 2.2 Gaussian stochastic process models

Gaussian stochastic process (GSP) or “Kriging” meta-models have proven useful for the modeling and optimization of expensive black-box functions (Matheron 1963; Santner et al. 2018). The derived empirical models can generate surrogate predictors,  $\mathcal{F}_f$  and  $G_j$ , for the objectives and constraint functions,  $f$  and  $g_j$ , in (1).

We assume that the decision vector  $\mathbf{x}$  has  $d$  dimensions, and  $n$  is the current total number of data points. The assumptions of GSP models are often expressed in terms of outputs,  $Y(\mathbf{x})$ , regression terms,  $h(\mathbf{x})$ , and random terms  $Z(\mathbf{x})$ . The assumed model is

$$Y(\mathbf{x}) = h(\mathbf{x}) + Z(\mathbf{x}). \tag{4}$$

Where  $Z(\mathbf{x})$  is a zero mean stationary Gaussian process with  $\operatorname{Cov}(Z(\mathbf{x}_i), Z(\mathbf{x}_j)) = \sigma^2 R_{i,j}(\theta, p)$ , where  $R_{i,j}(\theta, p)$  is defined in (5).

In our numerical examples, we make the common assumption that the regression terms are constant, i.e.,  $h(\mathbf{x}) = \beta_0$ . However, detrending data using regression can be critical in some problems.

The GSP model parameters are  $d$  dimensional vectors of correlation parameters,  $\theta$ , and exponents,  $p$ . A critical model quantity is the correlation matrix,  $\mathbf{R}$ , between the responses at the points,  $x_1, \dots, x_n$ . Here, we assume the standard Gaussian correlation function  $\mathbf{R}$  with  $i, j$ th entry  $R_{i,j}$ :

$$R_{i,j}(\theta, p) = \prod_{k=1}^m e^{-\theta_k (x_k - x_{i,k})^{p_k}}. \tag{5}$$

where  $R_{i,j}(\theta, p)$  is only a valid correlation function when  $0 < p_k \leq 2$  for all  $k$ . And the Gaussian correlation function assumes all  $p_k = 2$ .

The best linear unbiased predictor for the regression coefficients in  $h(\mathbf{x}) = \beta_0$  is

$$\beta_{\text{est}}(\theta, p) = (\mathbf{1}' \mathbf{R}^{-1} \mathbf{1})^{-1} \mathbf{1}' \mathbf{R}^{-1} \mathbf{y} \tag{6}$$

where  $\mathbf{1}$  is an  $n$ -dimensional vector of 1s. This coefficient is used for estimating the point-specific variance  $s_{\text{est}}^2(\mathbf{x} | \theta, p)$  and optimizing the likelihood. The standard formula to predict the variance of the GSP at point  $\mathbf{x}$  is

$$s_{\text{est}}^2(\mathbf{x} | \theta, p) = n^{-1} (\mathbf{y} - \mathbf{1} \beta_{\text{est}})' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1} \beta_{\text{est}}). \tag{7}$$

The values of  $\theta$  and  $p$  that maximize the right-hand side of (8) are the MLEs:

$$\max_{\theta, p} \ln L(\theta, p) = - \max_{\theta, p} \frac{n \ln s_{\text{est}}^2 + \ln[\det(\mathbf{R})]}{2}. \tag{8}$$

Predictions from the derived meta-model are then generated using the best linear unbiased predictor (BLUP, Santner et al. 2018):

$$\mathcal{F}(\mathbf{x}) = \beta_{\text{est}} + \mathbf{r}'(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\beta_{\text{est}}) \tag{9}$$

where  $\mathbf{r}(\mathbf{x}) = [R(\mathbf{x}, \mathbf{x}_1), \dots, R(\mathbf{x}, \mathbf{x}_n)]'$  with  $R(\mathbf{x}_i, \mathbf{x}_j)$  given by  $R(\mathbf{x}_i, \mathbf{x}_j) = \prod_{k=1}^m e^{-\theta_k(x_{i,k} - x_{j,k})^{p_k}}$ . The BLUP is a predictor with pleasing properties, including that it passes through the design points and provides a smooth interpolation between the points when  $p_k = 2$ . Far from the points (depending on the estimated correlation parameters,  $\theta$ ), the BLUP returns to the regression model. In our examples, this means that the model returns to a constant value.

### 2.3 Efficient global optimization and variants

The method proposed by ur Rehman and Langelaar (2017) to solve (1) is called robust EGO. Because the worst possible environmental conditions are not precisely known at any iteration for a given set of control variables,  $\mathbf{x}_c$ , the current best estimate for the objective value,  $\hat{u}_t$ , is needed. Also, the estimated worst possible objective value for a given set of environmental variables,  $\mathbf{x}_e$ , is needed  $\hat{l}_t$ . Formulas to estimate these quantities given the meta-models  $\mathcal{F}_f$  and  $\mathcal{G}_j$  are

$$\begin{aligned} \hat{u}_t &= \min_{\mathbf{x}_c \in \mathbb{X}_c} \mathcal{F}_f(\mathbf{x}_c, \mathbf{x}_{e,t}^*) | \mathcal{G}_j(\mathbf{x}_c, \mathbf{x}_{e,t}^*) \leq 0 \forall j \\ &= 1, \dots, q, \mathbf{x}_c \in \mathbb{X}_c, \mathbf{x}_e \in \mathbb{X}_e \end{aligned} \tag{10}$$

and

$$\begin{aligned} \hat{l}_t &= \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{F}_f(\mathbf{x}_{c,t}^*, \mathbf{x}_e) | \mathcal{G}_j(\mathbf{x}_{c,t}^*, \mathbf{x}_e) \leq 0 \forall j \\ &= 1, \dots, q, \mathbf{x}_c \in \mathbb{X}_c, \mathbf{x}_e \in \mathbb{X}_e. \end{aligned} \tag{11}$$

At iteration  $t$ , two quantities are successively estimated corresponding to the environmental conditions to study,  $\mathbf{x}_{e,t}^*$ , and control conditions to study,  $\mathbf{x}_{c,t}^*$ . Intermediate quantities are the improvement function,  $I_c(\mathbf{x}_c)$  at the point  $\mathbf{x}_c$ , which is defined as  $I_c(\mathbf{x}_c) = \max(\hat{u}_t - \mathcal{F}_f(\mathbf{x}_c, \mathbf{x}_{e,t}^*))$  and the constraint violation function,  $V_{j,c}(\mathbf{x}_c)$  at the point  $\mathbf{x}_c$ , which is defined as  $V_{j,c}(\mathbf{x}_c) = \max(\hat{g}_{j,\max}(\mathbf{x}_c) - 0)$ ,  $j = 1, \dots, q$ . Similar definitions hold for  $I_e(\mathbf{x}_e)$  and  $V_{j,e}(\mathbf{x}_e)$ ,  $j = 1, \dots, q$ . The so-called worst-case Kriging functions are  $\hat{f}_{\max}(\mathbf{x}_e) = \mathcal{F}_f(\mathbf{x}_{c,t}^*, \mathbf{x}_e)$ ,  $\hat{g}_{j,\max}(\mathbf{x}_c) = \mathcal{G}_j(\mathbf{x}_c, \mathbf{x}_{e,t}^*) \forall j = 1, \dots, q$ ,  $\hat{f}_{\max}(\mathbf{x}_e) = \mathcal{F}_f$

$(\mathbf{x}_{c,t}^*, \mathbf{x}_e)$ ,  $\hat{g}_{j,\max}(\mathbf{x}_c) = \mathcal{G}_j(\mathbf{x}_{c,t}^*, \mathbf{x}_e) \forall j = 1, \dots, q$ . Then, the intermediate quantities are

$$\begin{aligned} E[I_c(\mathbf{x}_c)] &= -(\hat{u}_t - \hat{f}_{\max}(\mathbf{x}_c)) \Phi\left(\frac{\hat{u}_t - \hat{f}_{\max}(\mathbf{x}_c)}{s(\mathbf{x}_c, \mathbf{x}_{e,t}^*)}\right) \\ &\quad - s(\mathbf{x}_c, \mathbf{x}_{e,t}^*) \phi\left(\frac{\hat{u}_t - \hat{f}_{\max}(\mathbf{x}_c)}{s(\mathbf{x}_c, \mathbf{x}_{e,t}^*)}\right) \end{aligned} \tag{12}$$

and

$$\begin{aligned} E[V_{j,c}(\mathbf{x}_c)] &= (\hat{g}_{j,\max}(\mathbf{x}_c) - 0) \Phi\left(\frac{\hat{g}_{j,\max}(\mathbf{x}_c) - 0}{s_j(\mathbf{x}_c, \mathbf{x}_{e,t}^*)}\right) \\ &\quad + s_j(\mathbf{x}_c, \mathbf{x}_{e,t}^*) \phi\left(\frac{\hat{g}_{j,\max}(\mathbf{x}_c) - 0}{s_j(\mathbf{x}_c, \mathbf{x}_{e,t}^*)}\right). \end{aligned}$$

$$\begin{aligned} E[I_e(\mathbf{x}_e)] &= (\hat{l}_t - \hat{f}_{\max}(\mathbf{x}_e)) \Phi\left(\frac{\hat{l}_t - \hat{f}_{\max}(\mathbf{x}_e)}{s(\mathbf{x}_{c,t}^*, \mathbf{x}_e)}\right) \\ &\quad + s(\mathbf{x}_{c,t}^*, \mathbf{x}_e) \phi\left(\frac{\hat{l}_t - \hat{f}_{\max}(\mathbf{x}_e)}{s(\mathbf{x}_{c,t}^*, \mathbf{x}_e)}\right) \end{aligned} \tag{13}$$

and

$$\begin{aligned} E[V_{j,e}(\mathbf{x}_e)] &= (\hat{g}_{j,\max}(\mathbf{x}_e) - 0) \Phi\left(\frac{\hat{g}_{j,\max}(\mathbf{x}_e) - 0}{s_j(\mathbf{x}_{c,t}^*, \mathbf{x}_e)}\right) \\ &\quad + s_j(\mathbf{x}_{c,t}^*, \mathbf{x}_e) \phi\left(\frac{\hat{g}_{j,\max}(\mathbf{x}_e) - 0}{s_j(\mathbf{x}_{c,t}^*, \mathbf{x}_e)}\right). \end{aligned}$$

The two expected improvement optimizations for infill candidate selections are

$$\mathbf{x}_{c,t}^* = \operatorname{argmin}_{\mathbf{x}_c \in \mathbb{X}_c} EI_{\text{cg}} = \operatorname{argmin}_{\mathbf{x}_c \in \mathbb{X}_c} \{E[I_c(\mathbf{x}_c)] | E[V_{j,c}(\mathbf{x}_c)]\}. \tag{14}$$

and

$$\mathbf{x}_{e,t}^* = \operatorname{argmax}_{\mathbf{x}_e \in \mathbb{X}_e} EI_{\text{eg}} = \operatorname{argmax}_{\mathbf{x}_e \in \mathbb{X}_e} \{E[I_e(\mathbf{x}_e)] | E[V_{j,e}(\mathbf{x}_e)]\}. \tag{15}$$

With these definitions, the robust EGO from ur Rehman and Langelaar 2017 is defined in Algorithm 1. Therefore, the method progressively identifies undesirable environmental conditions and desirable controllable parameters. The stopping conditions may relate to convergence, termination of a computational budget (as in our numerical examples), or the attainment of quality objectives.

**Algorithm 1:** Robust EGO

- 1 Initialize: Perform  $n_0$  experimental runs following a Latin Hypercube (LHC) array.
- 2 Solve Equation (8) to estimate the Kriging parameters.
- 3 Solve Equation (3) for  $\mathbf{x}_{c,t}^*$  with  $\mathbb{X}_e, \mathbb{X}_c$  temporarily the experimental runs;  $t = 0$ .
- 4 While: {Stopping criteria are not satisfied}
- 5  $t = t + 1$ ;
- 6 Solve Equation (10) and then Equation (14) for  $\mathbf{x}_{e,t}^*$  using  $\mathbf{x}_c = \mathbf{x}_{c,t-1}^*$ .
- 7 Solve Equation (11) and then Equation (15) for  $\mathbf{x}_{c,t}^*$  using  $\mathbf{x}_e = \mathbf{x}_{e,t}^*$ .
- 8 Perform an experimental run at the point  $(\mathbf{x}_{c,t}^*, \mathbf{x}_{e,t}^*)$ .
- 9 Solve Equation (8) with the full data to estimate the Kriging parameters.

Note that Algorithm 1 reduces to superEGO from Sasena et al. (2002) if  $|\mathbb{X}_c| = 1$  and the environmental variables are fixed. Then, step 6 is trivial, and step 7 is the constrained EI optimization. Also, if there are no constraints, i.e.,  $q = 0$ , then (12) reduces to the standard unconstrained EI formula (Jones et al. 1998). The reduction to standard unconstrained EGO is not exact because EGO originally used the best-obtained prediction value for  $\hat{r}_k$  rather than attempting to predict that value using the GSP model in (9). Here, we seek a further generalization of Algorithm 1 to include a tabu list for the infill optimization.

**2.4 Tabu search methods**

Tabu search (TS) is an evolving topic, but the core method elements persist (Glover 1986, 1989, 1990; Chelouah and Patrick 2000). An initial solution is generated, perhaps using a pre-procedure. Then, candidate solutions of interest are identified. The restrictions from an updating list of off-limits or “tabu” solutions are considered in selecting which candidate is to be sampled. If the sampled candidate offers an improvement, then it becomes the current best solution. Stopping criteria are checked. If the criteria are not met, then the restriction list is potentially updated, adding or removing items depending on the specific rules chosen.

Note that this structure is already similar to EGO methods. Both methods identify desirable candidates and take single steps, preserving their best solutions so far. The infill optimizations in EGO promise to provide a desirable way to identify and screen candidates. The major difference is the application of tabu lists during candidate screening. This is the difference that we seek to address in our proposed methods.

**2.5 The DIRECT method**

DIRECT is a derivative-free method for constrained (or unconstrained), non-linear optimization (Jones 2009b). DIRECT works by using Lipschitz theory to divide up the design space into hyper-rectangles. At each iteration, the set of hyper-

rectangles most likely to contain good function values are further subdivided.

DIRECT is used by superEGO and tabu EGO to find optimal Kriging model parameters and to solve the infill sampling criterion (ISC) sub-optimization problem.

**3 Proposed methods**

As mentioned previously, both EGO and tabu searches share a common structure of initialization followed by single steps leading to evaluations and iteration. A key difference for our proposed tabu EGO methods compared with other EGO methods relates to the infill optimizations being restricted by a list of “tabu” solutions. As for ordinary EGO, the expected improvement functions are successively optimized to obtain the corresponding environmental conditions,  $\mathbf{x}_{e,t}^*$ , and control conditions,  $\mathbf{x}_{c,t}^*$ . However, the closest relevant point,  $\mathbf{c}_t^*(\mathbf{x}_c)$  (which will be written as  $\mathbf{c}_t^*$  for simplification), on the tabu list is considered in a penalty term involving weights  $\mu_1$  and closeness  $\varepsilon$  as follows:

$$\begin{aligned} \mathbf{x}_{c,t}^* &= \operatorname{argmin}_{\mathbf{x}_c \in \mathbb{X}_c} EI_{cg} \\ &= \operatorname{argmin}_{\mathbf{x}_c \in \mathbb{X}_c} \{E[I_c(\mathbf{x}_c)]E[V_{j,c}(\mathbf{x}_c)] + \mu_1(\varepsilon - \min(d(\mathbf{x}_c, \mathbf{c}_t^*), \varepsilon))\}. \end{aligned} \tag{16}$$

where  $d(\mathbf{x}_c, \mathbf{c}_t^*) = \sqrt{\sum_{i=1}^N (\mathbf{x}_c - \mathbf{c}_{t,i}^*)^2}$ ,  $\mathbf{c}_t \in \mathbb{X}_{t,c}$ .

$$\begin{aligned} \hat{f}_{\max}(\mathbf{x}_c) &= \mathcal{F}_f(\mathbf{x}_c, \mathbf{x}_{e,t}^*); \hat{g}_{j,\max}(\mathbf{x}_c) = \mathcal{G}_j(\mathbf{x}_c, \mathbf{x}_{e,t}^*), \forall j \\ &= 1, \dots, q, \mathbf{x}_c \in \mathbb{X}_c, \mathbf{x}_e \in \mathbb{X}_e, \end{aligned}$$

$$\mu_1 = \frac{\hat{f}_{\max}(\mathbf{x}_c)}{\varepsilon}, \mathbb{X}_{t,c} \text{ is the tabu list for the control variables.}$$

Similarly, for the environmental variable optimization phase, we have



$$\begin{aligned} \mathbf{x}_{e,t}^* &= \operatorname{argmax}_{\mathbf{x}_e \in \mathbb{X}_e} EI_{eg} \\ &= \operatorname{argmax}_{\mathbf{x}_e \in \mathbb{X}_e} \{E[I_e(\mathbf{x}_e)]|E[V_{j,e}(\mathbf{x}_e)] + \mu_2(\varepsilon - \min(d(\mathbf{x}_e, \mathbf{e}_t^*), \varepsilon))\}. \end{aligned} \tag{17}$$

where  $d(\mathbf{x}_e, \mathbf{e}_t) = \sqrt{\sum_{i=1}^N (\mathbf{x}_e - \mathbf{e}_{t,i})^2}$ ,  $\mathbf{e}_t \in \mathbb{X}_{t,e}$ .

$$\begin{aligned} \hat{f}_{\max}(\mathbf{x}_e) &= \mathcal{F}_f(\mathbf{x}_{c,t}^*, \mathbf{x}_e); \hat{g}_{j,\max}(\mathbf{x}_e) = \mathcal{G}_j(\mathbf{x}_{c,t}^*, \mathbf{x}_e), \forall j \\ &= 1, \dots, q, \mathbf{x}_c \in \mathbb{X}_c, \mathbf{x}_e \in \mathbb{X}_e, \end{aligned}$$

$$\mu_2 = \frac{\hat{f}_{\max}(\mathbf{x}_e)}{\varepsilon}, \mathbb{X}_{t,e}$$

is the tabu list for the environmental variables.

Equations (16) and (17) are simple generalizations of the previous equations with the penalties included for points being similar to previous points. In terms of these equations, Algorithm 2 provides a detailed description of the proposed tabu EGO method for unconstrained, simple constrained, or robust design problems. In our examples, we use  $\varepsilon = \min_{\mathbf{x}_i, \mathbf{x}_j \in \chi} |\mathbf{x}_i - \mathbf{x}_j|$ ,  $i \neq j$ ,  $1 \leq i, j \leq N$ , the minimum possible distance between the points in the tabu list, for our tuning parameter. Random Latin hypercubes with  $n = 11d$  runs are applied by default.

**Algorithm 2:** Tabu EGO

- 1 Initialize: Perform  $n_0$  experimental runs following an LHC array.
- 2 Solve Equation (8) to estimate the Kriging parameters.
- 3 Solve Equation (3) for  $\mathbf{x}_{c,t}^*$  with  $\mathbb{X}_e, \mathbb{X}_c$  temporarily the experimental runs;  $t = 0$ .
- 4 Initialize the combined tabu list ( $\mathbb{X}_{tabu,c}, \mathbb{X}_{tabu,e}$ ) with the points in the LHC array.
- 5 While: {Stopping criteria are not satisfied}
- 6  $t = t + 1$ ;
- 7 Solve Equation (10) and then Equation (16) for  $\mathbf{x}_{e,t}^*$  using  $\mathbf{x}_c = \mathbf{x}_{c,t-1}^*$ .
- 8 Append  $\mathbf{x}_{e,t}^*$  to  $\mathbb{X}_{tabu,e}$  (update the environmental variable in tabu list).
- 9 Solve Equation (11) and then Equation (17) for  $\mathbf{x}_{c,t}^*$  using  $\mathbf{x}_e = \mathbf{x}_{e,t}^{**}$ .
- 10 Append  $\mathbf{x}_{c,t}^*$  to  $\mathbb{X}_{tabu,c}$  (update the control variable in tabu list).
- 11 Perform an experimental run at the point  $(\mathbf{x}_{c,t}^*, \mathbf{x}_{e,t}^*)$ .
- 12 If  $\{\hat{l}_\kappa < f(\mathbf{x}_{c,t}^*, \mathbf{x}_{e,t}^*) \text{ AND } g_j(\mathbf{x}_{c,t}^*, \mathbf{x}_{e,t}^*) \leq 0 \forall j\}$  Then  $\mathbf{x}_{e,t}^{**} = \mathbf{x}_{e,t}^*$ . (Update solution.)
- 13 If  $\{\mathbf{x}_{c,t}^*, \mathbf{x}_{e,t}^* \text{ AND } g_j(\mathbf{x}_{c,t}^*, \mathbf{x}_{e,t}^*) \leq 0 \forall j\}$  Then  $\mathbf{x}_{c,t}^{**} = \mathbf{x}_{c,t}^*$ . (Update worst.)
- 14 Solve Equation (8) with the full data to estimate the Kriging parameters.

It is noted that your algorithm only uses points at which the black-box function has been evaluated for purposes of estimating the optimum which is consistent with superEGO algorithm. Therefore, tabu EGO differs from other forms of EGO, including superEGO (Sasena et al. 2002) and robust EGO (ur Rehman and Langelaar 2017):

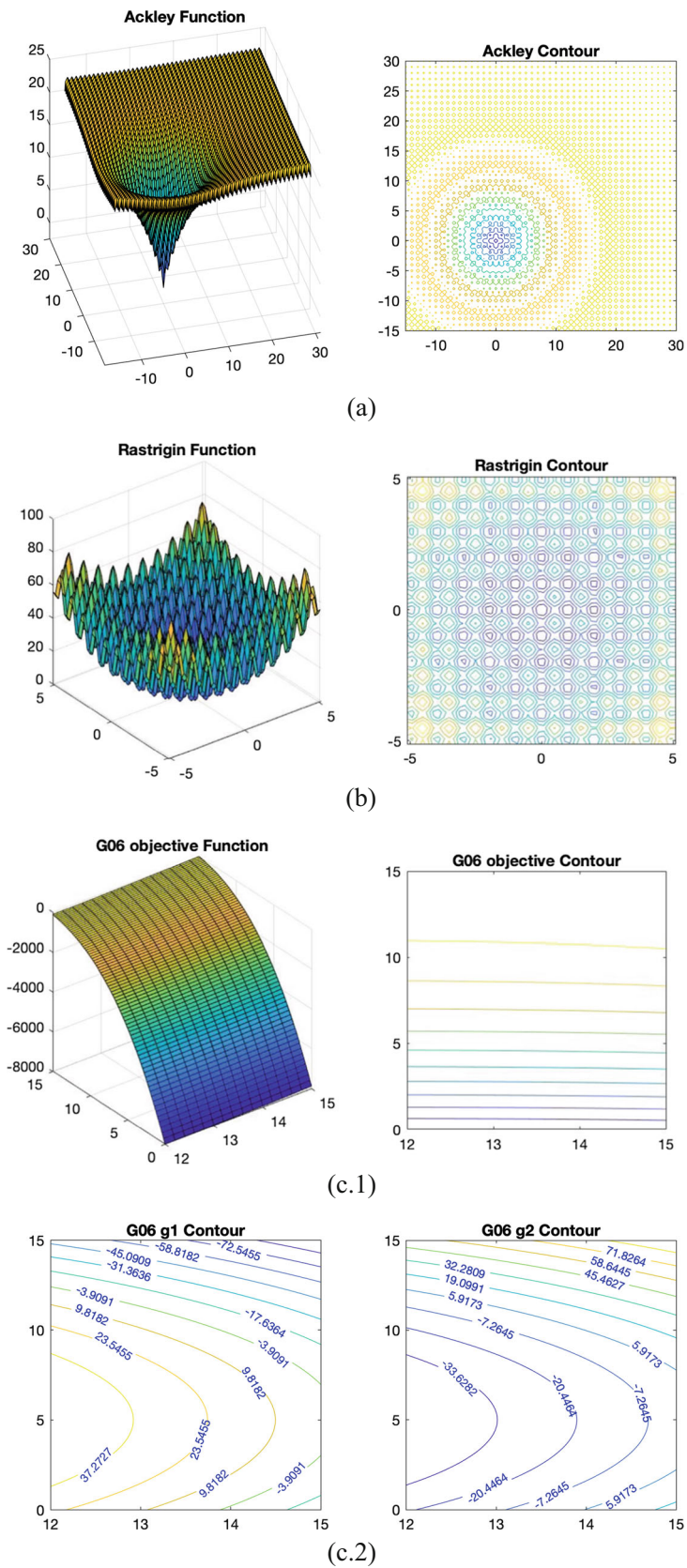
- A tabu list is included within the infill optimizations to screen candidates for novelty.
- Specifically, our penalty is either  $\mu_1\varepsilon/\mu_2\varepsilon$ , or a value that linearly decreases to zero depending on the sub-optimization problem as points differ from those on the tabu list.
- In addition, our method carefully updates and applies the apparent best solution,  $\mathbf{x}_{c,t}^{**}$ , and worst environmental condition,  $\mathbf{x}_{e,t}^{**}$  in lines (12) and (13). This does not directly

relate to the tabu search aspects, but we have found that it improves the method performance.

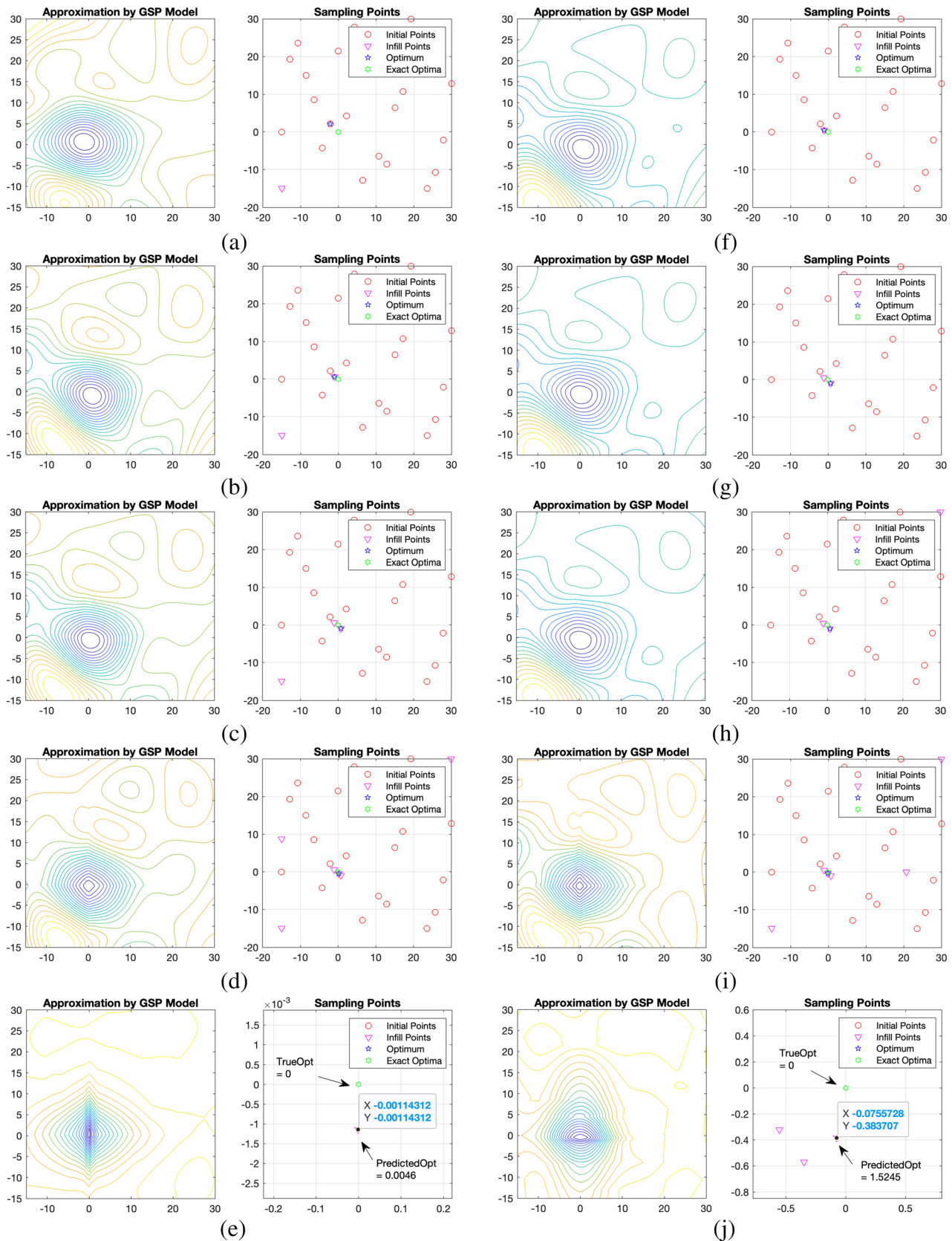
**4 Numerical comparisons**

**4.1 Illustrations of tabu EGO**

We illustrate the behavior of tabu EGO and its comparison to the traditional EGO method with the Ackley, Rastrigin, and G06 functions. Figure 1 shows the 3D and contour plots of Ackley, Rastrigin, and G06 benchmark functions. Both Ackley and Rastrigin functions pose a risk for optimization algorithms, particularly iterative algorithm algorithms, to be trapped in one of its many local minima. Note in Fig. 2 that, by employing the tabu search method, tabu EGO converges very



**Fig. 1** 3D and contour plots of Ackley, Rastrigin, and G06 benchmark functions. **a** Ackley, **b** Rastrigin, **(c.1)** G06 Objective function, **(c.2)** G06 Constraints function



**Fig. 2** Evolution of approximation of Ackley function (2 variables) with tabu EGO and superEGO. **a** Cycle 01 – tabu EGO, **b** Cycle 02 – tabu EGO, **c** Cycle 03 – tabu EGO, **d** Cycle 06 – tabu EGO, **e** Cycle 20 –

tabu EGO, **f** Cycle 01 – superEGO, **g** Cycle 02 – superEGO, **h** Cycle 03 – superEGO, **i** Cycle 06 – superEGO, **j** Cycle 20 – superEGO



quickly to the optimum with high accuracy (0.0046) for Ackley function at cycle 20, while superEGO obtained a minimum of 1.5245 when the maximum function evaluation reached. In addition, if we let tabu EGO to continue with the exploration, the meta-model quality is continuously improved.

We observed the same performance of tabu EGO for other functions as well. For instance is the evolution of the Rastrigin and G06 function in Figs. 3 and 4. In those cases, tabu EGO has found the optimum points within high accuracy at cycle 26 (0.0258) and cycle 27 ( $-6.36E+03$ ) compared to superEGO.

Based on these preliminary results with unconstrained and constrained functions, we can conclude that tabu EGO has a good performance on driving EGO algorithm, with a single infill point per cycle. Note as more infill points are added per cycle, the faster is the convergence to the global optimum (exploitation), and the quality improvement (predictability) of the meta-model is visioned. Future work could improve the convergence rate of tabu EGO by including multiple infill points method.

## 4.2 Comparative study

The test problems in Table 1 include the Ackley, Griewank, Rastrigin, Eggholder, Schaffer, Rosenbrock, Mishra's Bird function, and other standard global optimization benchmarks (Ma and Simon 2017). We also adapted the test functions G04, G09, and G09 from standard global optimization benchmarks (Ma and Simon 2017) to the robust constrained problem by setting controllable and environmental variables. Therefore, problems 1–5 are unconstrained, problems 6–10 are simply constrained, and problems 11–13 are robust design problems. The feasibility ratio of the constrained problems is generally small such that it is challenging for black-box methods to find optimal or even feasible solutions.

We used the same  $n = 11d$  runs from a Latin hypercube for all EGO methods in each test problem. As noted previously, the penalty parameter  $\varepsilon = \min_{\mathbf{x}_i, \mathbf{x}_j \in \chi} |\mathbf{x}_i - \mathbf{x}_j|$ ,  $i \neq j$ ,  $1 \leq i, j \leq N$ , was applied for tabu EGO implementations.

For the robust design problems (7–9), only robust EGO offers a valid peer method. Therefore, we only compare these results with robust EGO results. For the comparison, the stopping criterion for all methods was a fixed budget of runs. This is an excellent way to simulate engineering applications restricted by expensive simulation (e.g., FEA, CFD, and thermal simulation in additive manufacturing). We permitted  $n = 50$  evaluations for test problems 1–6,  $n = 100$  evaluations for test problems 7–8, and  $n = 150$  evaluations for test problem 9.

Tables 2 and 3 show the resulting solutions of mean, best, and standard deviation for each method and test problem. The best results are marked in bold. All the test problems are repeated 20 times with different DOE seeds. The sample means

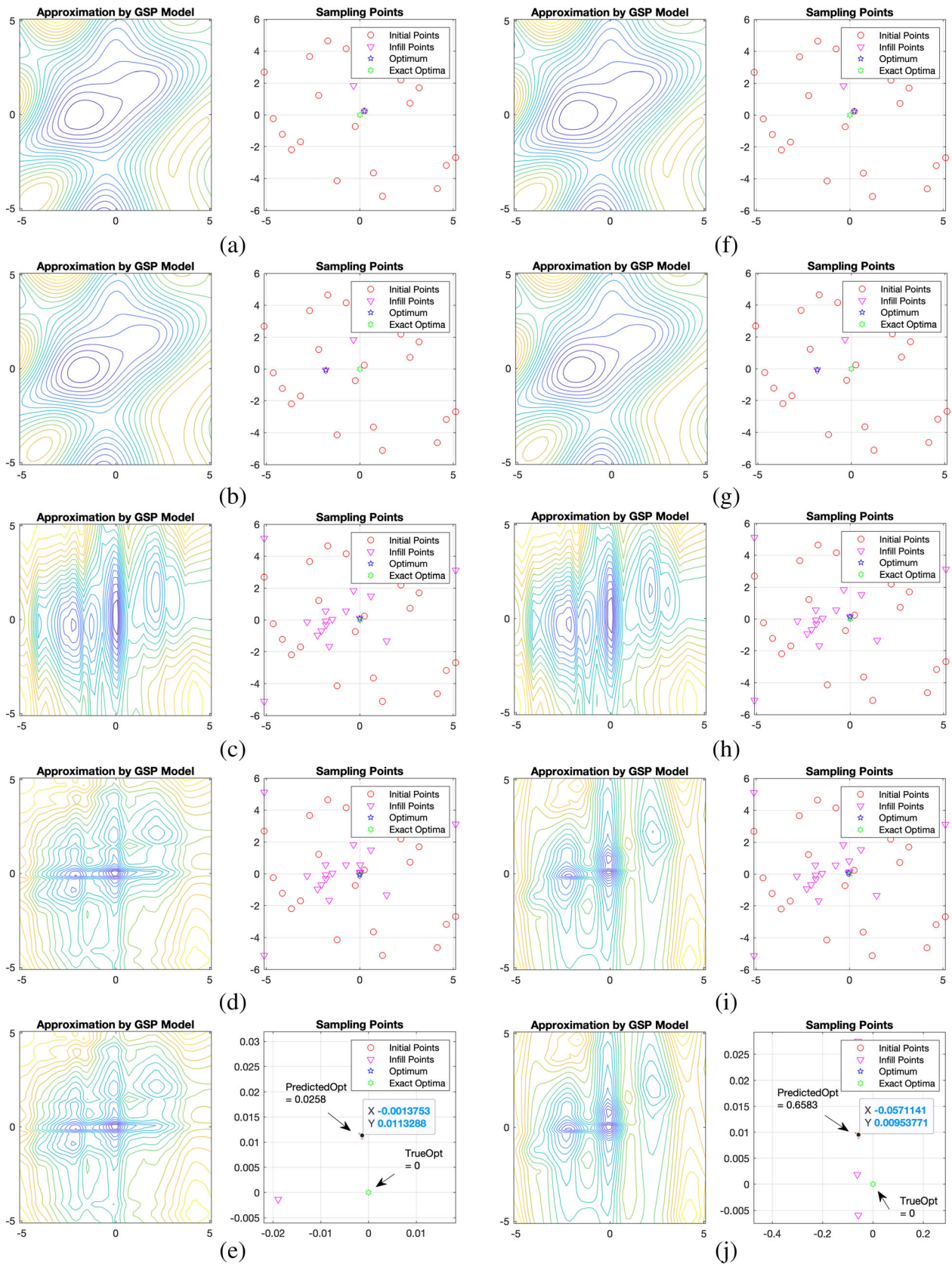
and standard deviations for comparison are provided. The DIRECT method does not require any experimental design or other random inputs, so the standard deviation is not fitted for this case.

In ten of the thirteen cases, tabu EGO obtains the best mean solution. For all the cases in which it does not, *fmincon* and superEGO get a better solution. In terms of best results over ten different runs, the tabu EGO algorithm wins in eight of the thirteen cases. Also, for the cases it does not, *fmincon* and TPE obtain a better one. The success rate (SR) of each algorithm for every test function is 1, except for test problems #7 and #13. For test problem #7, both tabu EGO and superEGO with constant regression got a 0.9 SR while superEGO with first-order regression got a 0.95 SR. This also suggests that the Kriging optimization method with higher-order regression could increase its success rate and algorithm performance (Forrester et al. 2006). However, except for test functions #2 and #9, tabu EGO tends to achieve superior results compared with the higher-order regression version of EGO. For test problem #13, the SR is 0.85 for robust EGO while SR is 1 for tabu EGO, which also implies tabu EGO has the potential of improving success rate by decrease the probability of returning to points in the tabu list.

Figure 5 presents the boxplots of the optimization results for twenty experimental designs for tabu EGO comparison with superEGO for the Ackley, Griewank, Rastrigin, and G06 test functions. Note that as the objective function is minimized as the function optimization cycles evolve, the dispersion is comparatively smaller in tabu EGO than superEGO.

In the same direction as the results presented by Viana et al. (2013), we found that by using tabu EGO with a penalty added in expected improvement function, a reduction in the dispersion of the results (minimum of the objective function) can be achieved as long as the optimization cycles evolve. For the Ackley case and Griewank case, the dispersion tends higher for tabu EGO than superEGO for the first 1–2 iterations. However, the dispersion for tabu EGO will be lower after 4 iterations for Ackley case and 3 iterations for Griewank case than superEGO as the algorithm evolves. This trend is more distinct for Griewank case than Ackley which maintain a slightly better dispersion from iterations 10–28 for tabu EGO than superEGO. For the Rastrigin and G06 function, the dispersion is similar for both tabu EGO and superEGO algorithm; however, the algorithm gets a better average result after 27 iterations for tabu EGO.

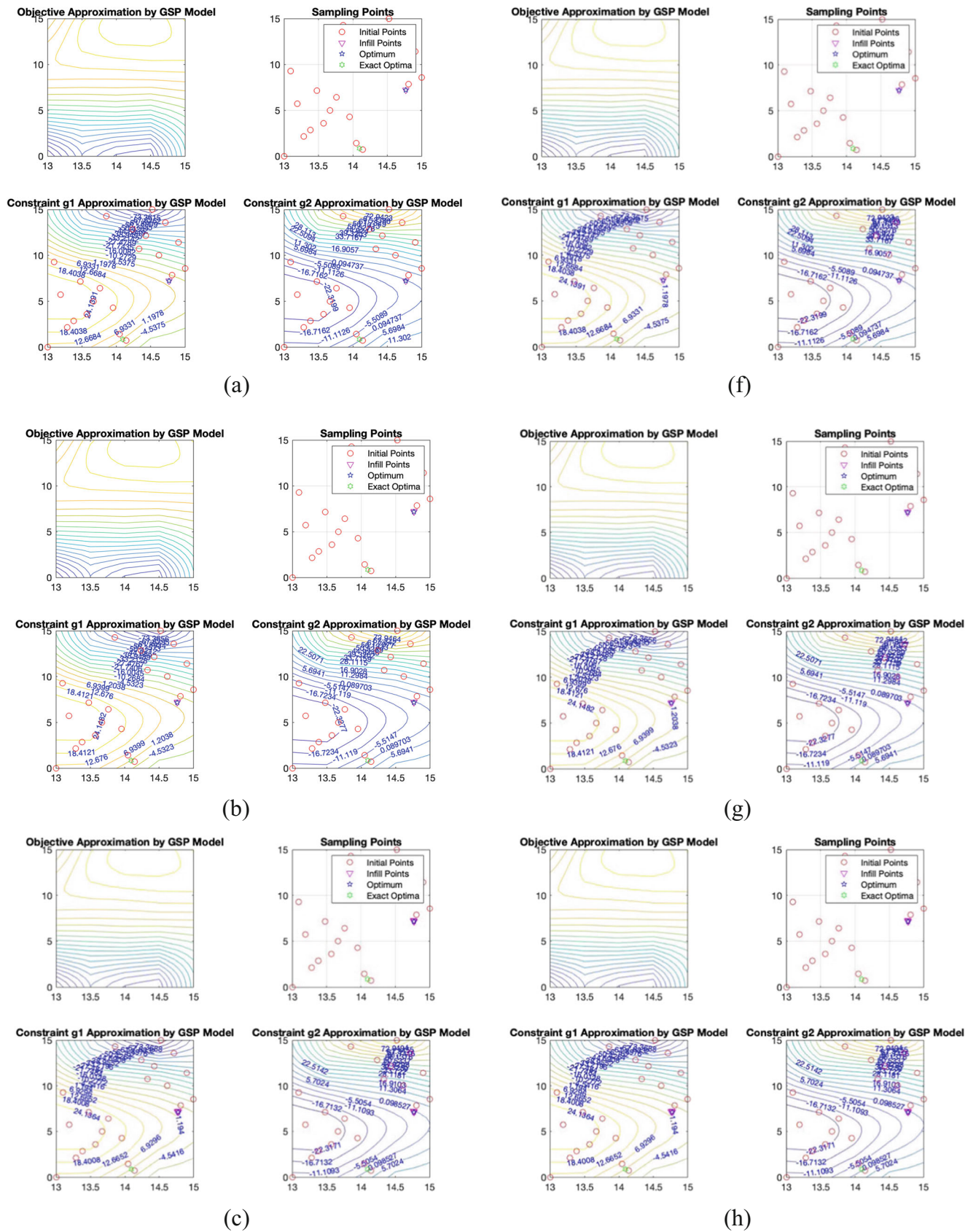
Figure 6 a-c present a comparison among tabu EGO and superEGO for Ackley, Griewank, and G06 functions. For both cases, the superEGO algorithm appears to outperform the tabu EGO algorithm except at cycle 23 for Ackley and Griewank function and the very last point for G06 function. Overall, the final performance of tabu EGO is better over all the alternative methods. For both cases, tabu EGO was faster in convergence rate than superEGO.



**Fig. 3** Evolution of approximation of Rastrigin function (2 variables) with tabu EGO and superEGO. **a** Cycle 01 – tabuEGO, **b** Cycle 02 – tabuEGO, **c** Cycle 16 – tabuEGO, **d** Cycle 20 – tabuEGO, **e** Cycle 26 –

tabuEGO, **f** Cycle 01 – superEGO, **g** Cycle 02 – superEGO, **h** Cycle 16 – superEGO, **i** Cycle 20 – superEGO, **j** Cycle 26 – superEGO





**Fig. 4** Evolution of approximation of G06 function (2 variables) with tabu EGO and superEGO. **a** Cycle 01 – tabuEGO, **b** Cycle 02 – tabuEGO, **c** Cycle 04 – tabuEGO, **d** Cycle 06 – tabuEGO, **e** Cycle 27 –

tabuEGO, **f** Cycle 01 – superEGO, **g** Cycle 02 – superEGO, **h** Cycle 04 – superEGO, **i** Cycle 06 – superEGO, **j** Cycle 27 – superEGO

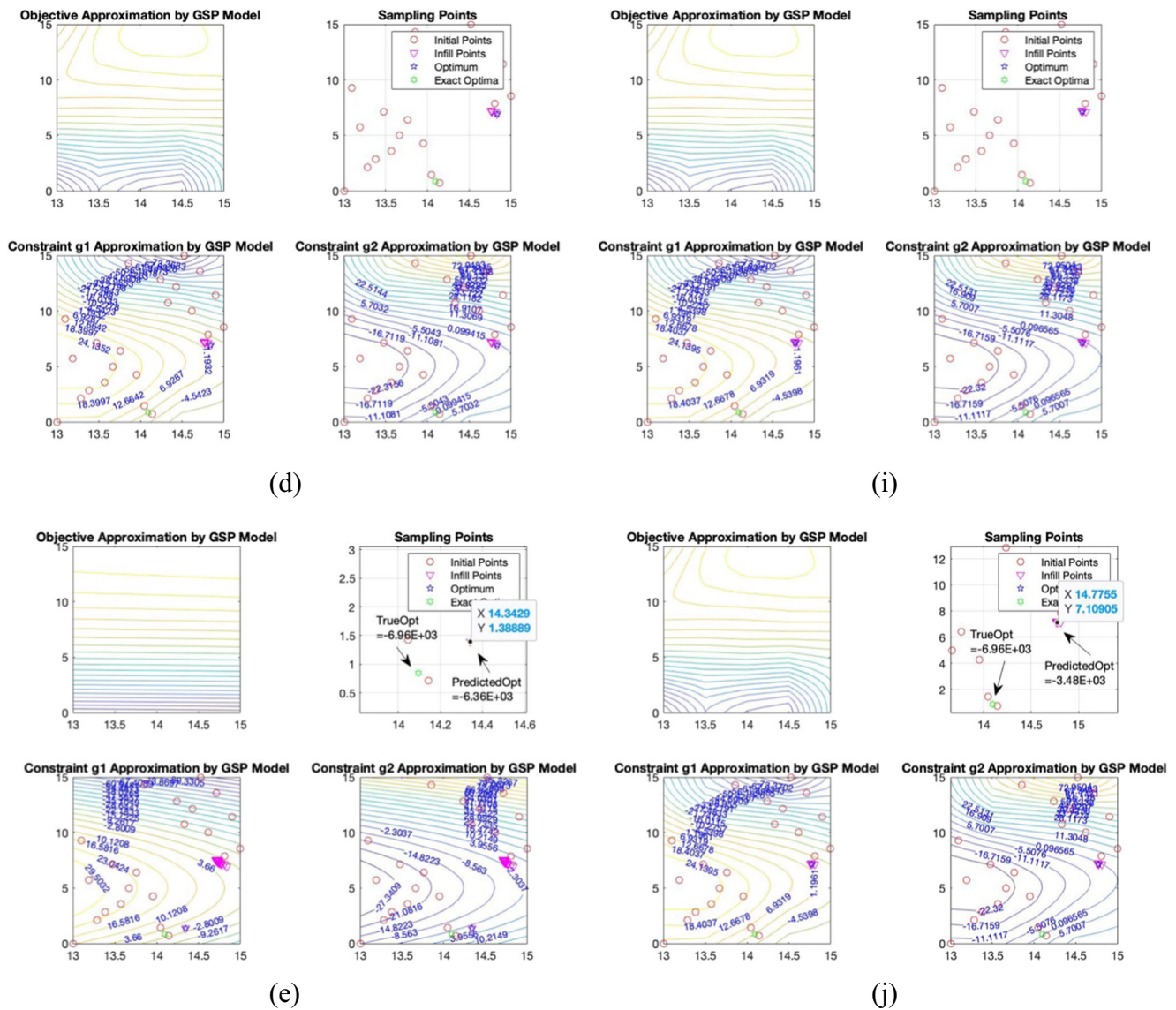


Fig. 4 (continued)

In order to compare the level of improvement ( $L_{imp}$ ) vs. optimization cycles, for all test problems, it was considered  $f_{eval}^{max} = 50$ . In this case,  $L_{imp}$  is calculated for each cycle  $k$  as

$$L_{imp} = 100\% \times \left| \frac{y_{min}^k - y_{min}^0}{y_{min}^{exact} - y_{min}^0} \right| \quad (18)$$

where  $y_{min}^0$  is the minimum for the initial sample,  $y_{min}^k$  is the minimum value at cycle  $k$ , and  $y_{min}^{exact}$  is the exact minimum value. Figure 6 d-f are to compare tabu EGO and superEGO in terms of computational costs, measured by the number of optimization cycles (function evaluations) required for the same level of improvement on the objective function at each optimization cycle.

As shown in the curves of Fig. 6d-f, we can observe that for Ackley and Griewank at the beginning of the optimization cycles, superEGO has lower cost in terms of the number of function evaluations than tabu EGO for the same level of improvement, but at some point (e.g., around 91% for Ackley and 95% for Griewank), the situation is more favorable to tabu EGO. In the case of the G06 function, the computational cost of both methods is quite similar before 55% of  $L_{imp}$  while tabu EGO outperforms superEGO when  $L_{imp}$  is around 67%. These results confirm the fact that tabu EGO, in general, is beneficial in terms of delivering results in reasonable processing time.

Note that additional experimental work is done on the effect of the initial sample size for the algorithm's results, and the results are summarized in Table 4. The results indicated



**Table 1** Test problems for the numerical comparisons

Name	Function descriptions	Source
#1	$n=2$ $\min_{x_1, x_2} f(x_1, x_2) = 20 + e - 20e^{-0.2 \sum_{i=1}^2 \frac{x_i^2}{n} - e^{\sum_{i=1}^2 \frac{\cos(2\pi x_i)}{n}}}$ $-15 \leq x_i \leq 30 \quad i = 1, 2.$ $\mathbf{x}^* = (0, 0), f^* = 0.$	Ackley
#2	$n=2$ $\min_{x_1, x_2} f(x_1, x_2) = \sum_{i=1}^2 \frac{x_i^2}{4000} - \prod_{i=1}^2 \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ $-600 \leq x_i \leq 600 \quad i = 1, 2.$ $\mathbf{x}^* = (0, 0), f^* = 0.$	Griewank
#3	$n=2$ $\min_{x_1, x_2} f(x_1, x_2) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$ $-5.12 \leq x_i \leq 5.12 \quad i = 1, 2.$ $\mathbf{x}^* = (0, 0), f^* = 0.$	Rastrigin
#4	$n=2$ $\min_{x_1, x_2} f(x_1, x_2) = -(x_2 + 47) \sin\left(\sqrt{\left x_2 + \frac{x_1}{2} + 47\right }\right) - x_1 \sin\left(\sqrt{ x_1 - (x_2 + 47) }\right)$ $-512 \leq x_i \leq 512 \quad i = 1, 2.$ $\mathbf{x}^* = (512, 404.2319), f^* = -959.6407.$	Eggholder
#5	$n=2$ $\min_{x_1, x_2} f(x_1, x_2) = 0.5 + \frac{\cos^2[\sin( x_1^2 - x_2^2 )] - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$ $-100 \leq x_i \leq 100 \quad i = 1, 2.$ $\mathbf{x}^* = (0, \pm 1.25313), f^* = 0.292579.$	Schaffer
#6	$n=2$ $\min_{x_1, x_2} f(x_1, x_2) = (x_1 - 10)^3 + (x_2 - 20)^3 + 3$ $s. t. : g_1(x_1, x_2) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100$ $g_2(x_1, x_2) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81$ $13 \leq x_1 \leq 15$ $0 \leq x_2 \leq 15$ $g_1(x_1, x_2) \leq 0; g_2(x_1, x_2) \leq 0.$ $\mathbf{x}^* = (14.095, 0.84296), f^* = -6961.8139.$	G06
#7	$n=2$ $\min_{x_1, x_2} f(x_1, x_2) = -\frac{\sin^3(2\pi x_1) \sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$ $s. t. : g_1(x_1, x_2) = x_1^2 - x_2^2 + 1$ $g_2(x_1, x_2) = 1 - x_1 + (x_2 - 4)^2$ $0 \leq x_i \leq 10 \quad i = 1, 2$ $g_1(x_1, x_2) \leq 0; g_2(x_1, x_2) \leq 0.$ $\mathbf{x}^* = (1.2279713, 4.2453733), f^* = -0.095825.$	G08
#8	$n=2$ $\min_{x_1, x_2} f(x_1, x_2) = -(1 - x_x)^2 + 100(x_2 - x_1^2)^2$ $s. t. : g_1(x_1, x_2) = (x_1 - 1)^3 - x_2 + 1$ $g_2(x_1, x_2) = x_1 + x_2 - 2$ $-1.5 \leq x_1 \leq 1.5, -0.5 \leq x_2 \leq 2.5$ $g_1(x_1, x_2) \leq 0; g_2(x_1, x_2) \leq 0.$ $\mathbf{x}^* = (1.0, 1.0), f^* = 0.$	Rosenbrock function constrained with a cubic and a line
#9	$n=2$ $\min_{x_1, x_2} f(x_1, x_2) = \sin(x_2) e^{[(1 - \cos x_1)^2]} + \cos(x_x) e^{[(1 - \sin x_2)^2]} + (x_1 - x_2)^2$ $s. t. : g(x_1, x_2) = (x_1 + 5)^2 + (x_2 + 5)^2$ $-10 \leq x_1 \leq 0, -6.5 \leq x_2 \leq 0.$ $g_1(x_1, x_2) \leq 0; g_2(x_1, x_2) \leq 0.$ $\mathbf{x}^* = (-3.1302468, -1.5821422), f^* = -106.7645367.$	Mishra's Bird function - constrained
#10	The three-bar truss design problem is to minimize the volume subject to stress constraints. This problem has two design variables and three constraints, and is formulated as	Three-bar truss design

**Table 1** (continued)

Name	Function descriptions	Source
#11	$n=2$ $f(x) = (2\sqrt{2}x_1 + x_2) \times l,$ $g_1(x) = \frac{(2\sqrt{2}x_1 + x_2)}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0,$ $g_2(x) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0,$ $g_3(x) = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0,$ $0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, l = 100\text{cm},$ $P = 2\text{KN/cm}^2, \sigma = 2\text{KN/cm}^2$ $f^* = 263.8958$	G04
#12	$n=5$ $\min_{x_1, \dots, x_5} f(x_1, \dots, x_5) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$ $s. t. : g_1(x_1, \dots, x_5) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92;$ $g_2(x_1, \dots, x_5) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5;$ $g_3(x_1, \dots, x_5) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_4 + 0.0021813x_3x_5 - 110;$ $g_4(x_1, \dots, x_5) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_4 - 0.0021813x_3x_5 + 90;$ $g_5(x_1, \dots, x_5) = 9.300961 + 0.0047026x_2x_5 + 0.0012547x_1x_4 + 0.0019085x_3x_5 - 25;$ $g_6(x_1, \dots, x_5) = -9.300961 - 0.0047026x_2x_5 - 0.0012547x_1x_4 - 0.0019085x_3x_5 + 20;$ $\max_{x_3, x_4} g(x_1, \dots, x_5) \leq 0$ Control variables: $x_1, x_2, x_3.$ Environmental variables: $x_4, x_5$ $78 \leq x_1 \leq 102$ $33 \leq x_2 \leq 45$ $27 \leq x_i \leq 45 \quad i = 3, 4, 5.$ $f^* = -30,453.0000$	G12
#13	$n=3$ $\min_{x_1, x_2} f(x_1, x_2, x_3) = -\frac{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^3}{100}$ $s. t. : g(x_1, x_2, x_3) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625$ $\max_{x_2, x_3} g(x_1, x_2, x_3) \leq 0, p, q, r = 1, 2, \dots, 9$ Control variables: $x_1$ Environmental variables: $x_2, x_3$ $0 \leq x_i \leq 10 \quad i = 1, 2, 3.$ $f^* = -1.0000$	G09
	$n=7$ $\min_{x_1, \dots, x_7} f(x_1, \dots, x_7) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$ $s. t. : g_1(x_1, \dots, x_7) = -127 + 2x_1^2 + 3x_4^4 + x_3 + 4x_4^2 + 5x_5;$ $g_2(x_1, \dots, x_7) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5;$ $g_3(x_1, \dots, x_7) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7;$ $g_4(x_1, \dots, x_7) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7;$ $\max_{x_4, \dots, x_7} g(x_1, \dots, x_7) \leq 0$ Control variables: $x_1, x_2, x_3$ Environmental variables: $x_4, \dots, x_7$ $-10 \leq x_i \leq 10 \quad i = 1, \dots, 7.$ $f^* = -822.5900$	

that the performance of the tabu EGO does not always enhance as the number of initial samples increases. Specifically, the algorithm got its best results for problem #2 when  $n = 8d$ , problem #3 when  $n = 8d$ , problem #4 when  $n = 10d$ , and problem #6 when  $n = 16d$ . For problems #1 and #5, the algorithm results are optimal when  $n = 11d$ . However, the performance of tabu EGO is not monotonously increased as

the initial sample size increased those two cases, which obtain their second better results for  $n = 6d$  (problem #1) and  $n = 8d$  (problem #6).

By changing the stopping criterion to relative error  $E =$

$$\left| \frac{\widehat{f}_{\min} - f_{\min}}{f_{\min}} \right|$$

and set  $E < 0.01$  for tabu EGO, we can compare

**Table 2** Numerical test results for solutions after 50 evaluations for test problems 1–6. The responses and design variables are dimensionless except for problem 10. Problem 10 has outputs in centimeters<sup>3</sup> and inputs in centimeters

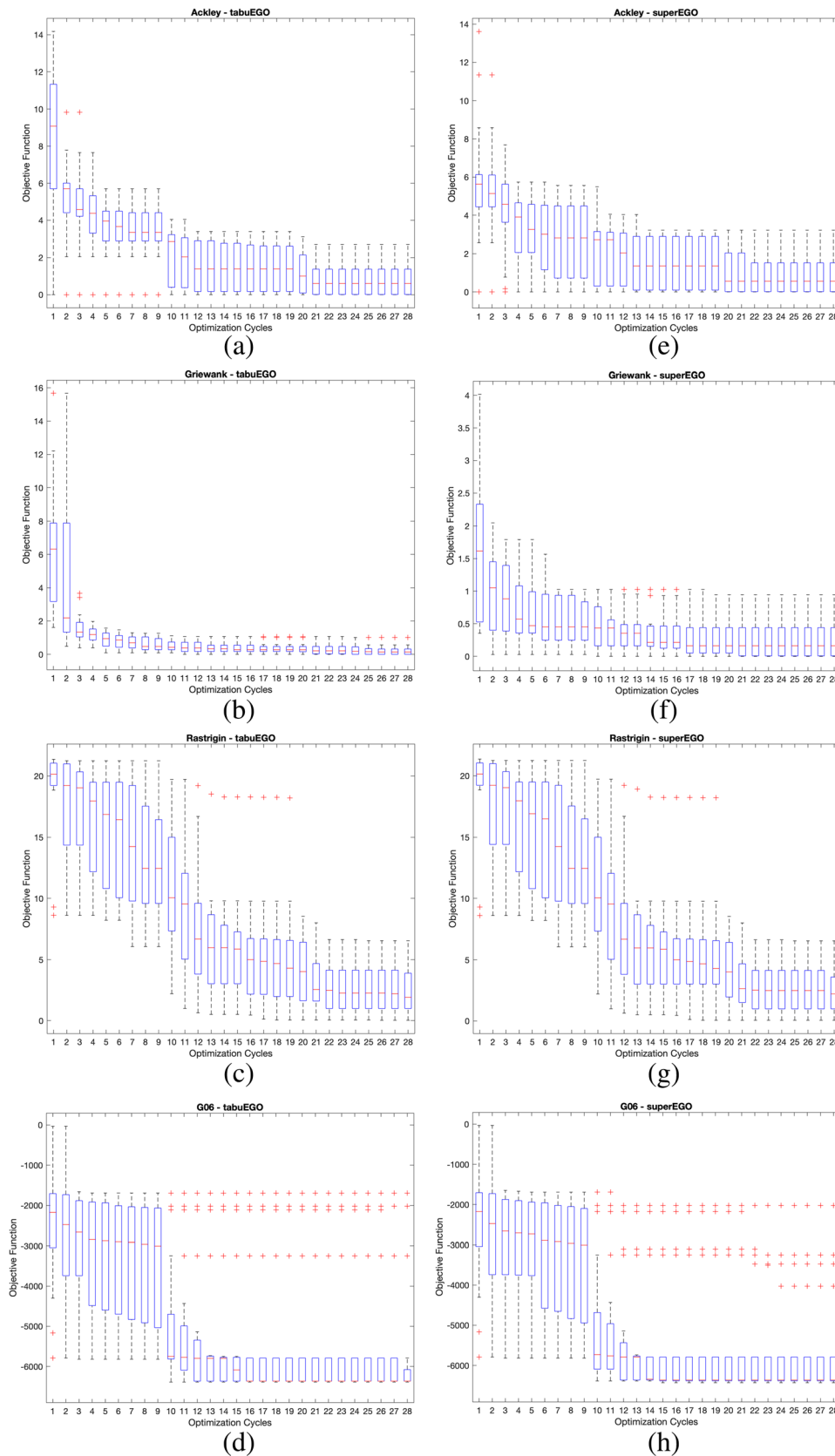
Prob.	Criteria	tabu EGO	superEGOk0	superEGOk1	DIRECT	fmincon	TPE
#1	Mean	<b>0.77521</b>	0.82716802	0.88772	N/A	10.6318412	1.99509532
	STD	1.00911728	0.88183514	0.74176529	N/A	4.72061935	<b>0.34528959</b>
	Best	<b>7.7435E-08</b>	5.1945E-07	0.0266	8.1617	2.57993055	1.1393235
#2	Mean	0.211615	0.224145	<b>0.18935</b>	N/A	49.5914514	1.64855025
	STD	0.34208793	<b>0.1910219</b>	0.28715915	N/A	24.5932104	0.88394953
	Best	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.0000</b>	2.07897736	0.57027292
#3	Mean	<b>2.16519</b>	2.240585	2.231575	N/A	17.9091651	6.05205471
	Std	2.10533683	2.11235791	<b>1.25534861</b>	N/A	8.70476054	3.09686107
	Best	0.0538	0.0536	0.0021	<b>0</b>	8.9546051	2.84578444
#4	Mean	<b>-751.43364</b>	-712.72311	-744.01392	N/A	-566.02319	-659.0315
	STD	147.501478	167.725313	141.135239	N/A	194.310591	115.695612
	Best	<b>-920.5677</b>	-883.9833	-881.2731	-724.8978	-753.05015	-869.82201
#5	Mean	<b>0.305735</b>	0.30591	0.3115	N/A	0.49667675	0.39354921
	STD	0.00684648	0.00755119	0.00810573	N/A	<b>0.0007163</b>	0.06057949
	Best	<b>0.2956</b>	<b>0.2956</b>	0.3021	0.4846	0.49606902	0.31930608
#6	Mean	<b>-5.70E+03</b>	-5.65E+03	-5.65E+03	N/A	-3.81E+03	N/A
	STD	983.065292	988.679946	1525.92985	N/A	<b>0.05567482</b>	N/A
	Best	-6387.1	<b>-6430.7</b>	-6430.5	-5254.5	-3249.5643	N/A
#7	Mean	<b>-6.96E-02</b>	-6.64E-02	-5.88E-02	N/A	-0.0557482	N/A
	STD	0.02684105	0.02723288	0.03088459	N/A	<b>0.01918962</b>	N/A
	Best	-0.0949	<b>-0.0955</b>	-0.0953	-0.023	-0.0898234	N/A
#8	Mean	<b>0.200425</b>	0.20582331	0.52978306	N/A	1.65206798	N/A
	STD	<b>0.41126541</b>	0.53333019	0.6826115	N/A	2.36187796	N/A
	Best	<b>0</b>	<b>0</b>	<b>0</b>	1.7864	4.3641E-07	N/A
#9	Mean	-106.67522	<b>-106.73024</b>	-106.67898	N/A	-77.268406	N/A
	STD	0.22924671	<b>0.03924444</b>	0.12534214	N/A	48.19548	N/A
	Best	<b>-106.7645</b>	-106.7643	<b>-106.7645</b>	-106.5496	-106.76454	N/A
#10	Mean	2.74304	2.760115	2.74304	N/A	<b>2.63970174</b>	N/A
	Std	0.09648001	0.07631617	0.09648001	N/A	<b>0.00468045</b>	N/A
	Best	2.6432	2.6432	2.6432	2.653	<b>2.63895851</b>	N/A

the results of G06, G08, and three-bar truss function to the results from eDIRECT-C method (Liu et al. 2017). It can see

that the tabu EGO has the potential of saving function runs as suggested by G08 function, which saves more than 50 runs to

**Table 3** Numerical test results for solutions after 100 evaluations for test problems 7–8, and 150 evaluations for test problem 9. The responses and design variables are dimensionless

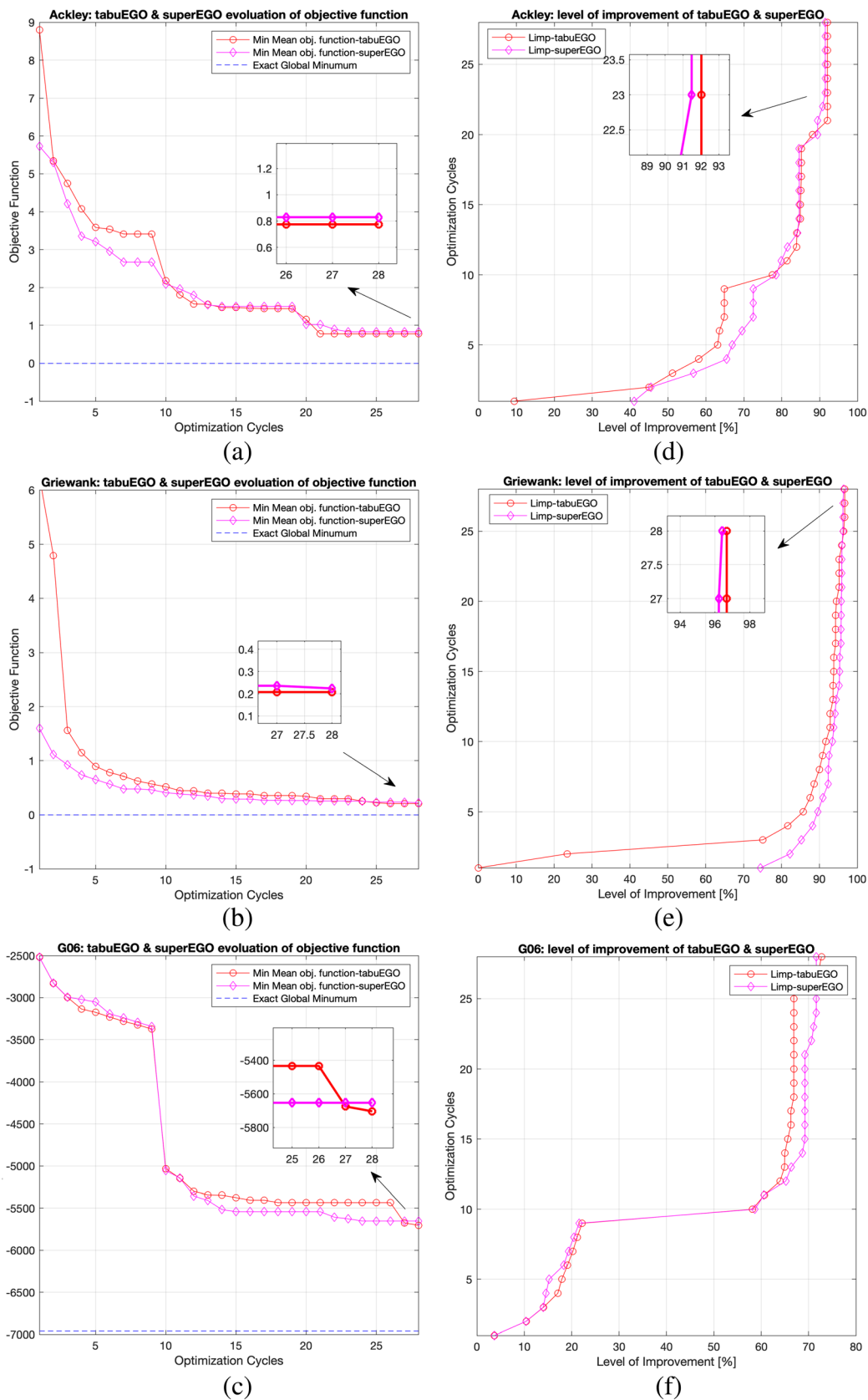
Prob.	Criteria	tabu EGO	robust EGO
#11	Mean	<b>-30,035.65</b>	-29,634.3
	Best	-30,453	<b>-30,589</b>
	Std	<b>276.716133</b>	490.9842
#12	Mean	<b>-0.969006</b>	-0.9407405
	Best	<b>-0.99999</b>	-0.9939
	Std	0.04240601	<b>0.03750872</b>
#13	Mean	<b>1463.73311</b>	409,895.602
	Best	<b>822.6929</b>	836.1874
	Std	<b>1665.95622</b>	1,176,664.78



**Fig. 5** Comparison of the convergence of tabu EGO (left boxplots) vs. superEGO (right boxplots), over 10 different initial DOE in each case for the functions Ackley, Griewank, Rastrigin, and G06. **a** Ackley, tabuEGO,

**b** Griewank, tabuEGO, **c** Rastrigin, tabuEGO, **d** G06, tabuEGO, **e** Ackley, superEGO, **f** Griewank, superEGO, **g** Rastrigin, superEGO, **h** G06, superEGO





**Fig. 6** Comparison of tabu EGO vs. other alternatives on the averaged results (a-c) and level of improvement (d-f) in each iteration. **a** Ackley, **b** Griewank, **c** G06, **d** Ackley, **e** Griewank, **f** G06

**Table 4** Results of tabu EGO for test functions #1-#6 based on different LHC design variables

Prob.	Criteria	tabu EGO, $n=4d$	tabu EGO, $n=6d$	tabu EGO, $n=8d$	tabu EGO, $n=10d$	tabu EGO, $n=11d$
#1	Mean	0.8245	0.4743	0.5882	0.5130	<b>0.4531</b>
	Std	0.8614	0.4237	0.4992	<b>0.3852</b>	0.5001
	Best	0.0005	2.3231E-07	<b>8.8818E-16</b>	0.0046	<b>8.8818E-16</b>
#2	Mean	0.1773	1.1909	<b>0.0967</b>	1.0199	0.1384
	Std	0.1482	2.1987	<b>0.1127</b>	1.5165	0.1451
	Best	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>
#3	Mean	<b>1.0295</b>	4.4539	1.7586	2.6868	2.2978
	Std	<b>0.9372</b>	4.0158	1.4796	2.3373	1.3329
	Best	<b>0.0102</b>	1.0235	0.0069	0.0154	0.0258
#4	Mean	-5642.2000	-5911.7300	-4666.3300	<b>-6256.5800</b>	-5407.8400
	Std	478.1101	348.0901	3681.7573	<b>245.6329</b>	1881.4660
	Best	-5811.7000	-6365.1000	<b>-6425.4000</b>	-6386.8000	-6384.0000
#5	Mean	2.7443	2.9161	2.7262	2.7624	<b>2.7166</b>
	Std	<b>0.0965</b>	0.3556	0.1056	0.1049	0.1028
	Best	2.653	2.6488	<b>2.645</b>	2.6635	2.6509
#6	Mean	-0.0731	<b>-0.0748</b>	-0.0704	-0.0669	-0.0743
	Std	0.0242	<b>0.0192</b>	0.0280	0.0252	0.0251
	Best	-0.0949	<b>-0.0958</b>	-0.0957	-0.0949	<b>-0.0958</b>

get a result with an error around 0.01 (Table 5). For the three-bar truss case, the algorithm requires slightly large function runs than eDIRECT-C but better than all the other methods while maintaining a high level of accuracy. However, for G06 test function, the result from tabu EGO got stuck by a local minimum, meaning that tabu EGO has the threat of preventing the algorithm from finding the exact minimum. This issue will be discussed in detail in Section 6.

## 5 Additive manufacturing case studies

Additive manufacturing (AM) or 3D printing has become one of the most promising commercial manufacturing techniques.

Benefits of AM include delivery savings as parts are made on-site, lower environmental wastes with less warehousing, and (in some cases) improved product quality and performance. The principal barriers of AM include cost, speed, quality, range of materials, and size limitations (Frazier 2014). Here, we focus on additive manufacturing process design with the objective of reducing costs and increasing the speed with quality constraints.

For our case studies, the Green's function thermal simulation method is used for evaluating the part quality constraints at different points in the  $X$ - $Y$  space (Schwalbach et al. 2019). The thermal simulations identify a lattice of source and measurement points and estimate the temperature histories. The parameter  $T_0$  is the initial temperature,  $\sigma$  is the physical size

**Table 5** Simple constrained comparison based on a 1% stopping criterion. In the table,  $f_{\min}$  is the minimum objective value. The average number of evaluations and the success rate (SR) are estimated using 10 problem instances

Prob.	Criteria	tabu EGO	eDIRECT-C	constrLMSRBF	CORBA	CiMPS	DIRECT
G06	$f_{\min}$	-6371.4100	<b>-6956.9684</b>	-6935.3844	-6961.8139	-6961.8139	N/F
	Avg. #evals.	44.5	<b>34.8</b>	>943.4	>1000.0	29,721.0	>1000.0
	SR	1.0	<b>1.0</b>	0.1	0.0	N/A	1.0
G08	$f_{\min}$	-0.095604	<b>-0.095822</b>	-0.088963	-0.095610	-0.095825	-0.095825
	Avg. #evals.	<b>105.1</b>	154.2	>197.2	980.6	11,914.0	326.0
	SR	0.8	<b>1.0</b>	0.9	0.2	N/F	1.0
Three-bar truss	$f_{\min}$	267.6860	<b>263.8958</b>	263.9168	263.9150	N/F	N/F
	Avg. #evals.	<b>51.1</b>	24.8	89.4	>486.2	N/F	>1000
	SR	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.8	N/F	0.000

over which the energy is deposited,  $\tau_i$  is the actual power deposit time,  $p_i$  is the laser power at point  $i$ ,  $\rho$  is the power (Ti-6Al-4 V) density, and  $\alpha$  is the thermal diffusivity. The temperature field is expressed by

$$T(r_j, t) = T_0 + \sum_i^N C_i \Theta(t - \tau_i) \lambda_i^{-3/2} e^{-\left(\frac{R_{ij}^2}{2\lambda_i}\right)} \quad (19)$$

where

$$C_i = \frac{p_i \Delta t}{\rho c_p \sqrt{2\pi\lambda_i}}, \lambda_i = \sigma^2 + 2\alpha(t - \tau_i), R_{ij} = |\vec{r}_j - \vec{r}_i| = \sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2}, \vec{r}_i = \langle x_i, y_i, z_i \rangle, \vec{r}_j = \langle x_j, y_j, z_j \rangle,$$

and  $\alpha = \frac{k}{\rho c_p}$ .

Figure 7 shows the thermal history at an arbitrary point in the  $X$ - $Y$  space for a given layer and scan pattern. The temperature reaches a maximum when the laser is closest to the point.

When the thermal temperature is estimated as the function time, two quality measures can be calculated: (1) times molten  $n_m$ , i.e., the number of times a discrete point is melted during the laser scanning process and (2) molten duration  $t_m$ , i.e., the total time a discrete point melted during the laser scanning process.

The thermal simulations based on (18) differ from many types of finite element methods in that there is a considerable incremental cost to obtaining results from each point in  $X$ - $Y$  space. Finite element methods (FEM) generally provide results across spatial points such as latent stresses or part distortion values, making it relatively easy to identify the worst points on the parts involved. In our simulations, however, the most difficult points on the part must be identified using a spatial optimization. Therefore, (19) involves speeding up

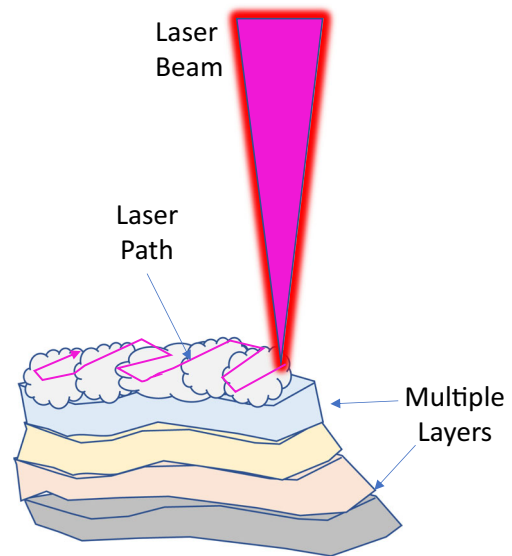


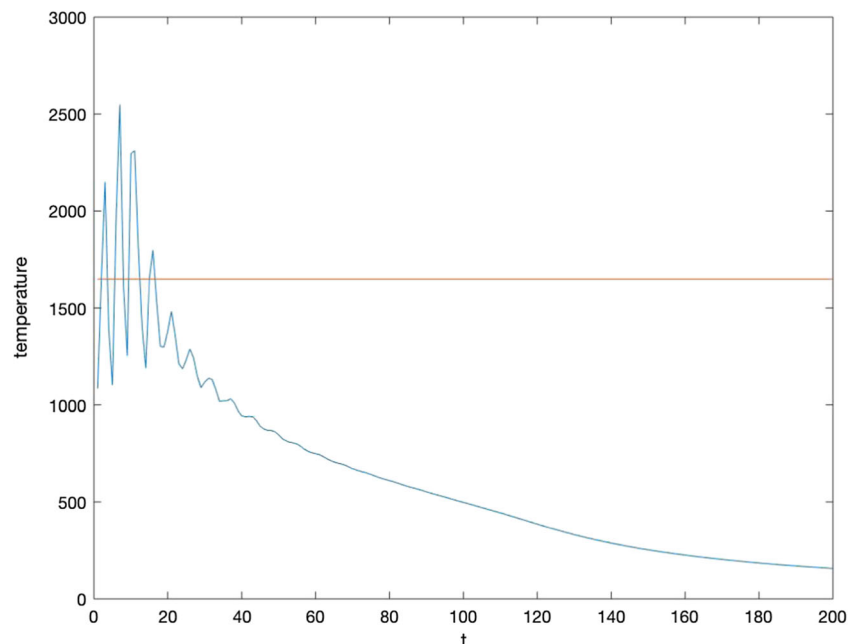
Fig. 8 Hypothetical laser path in additive manufacturing

the laser path as shown in Fig. 8. The parameters in (19) determine the raster pattern of the beam as indicated in Fig. 9.

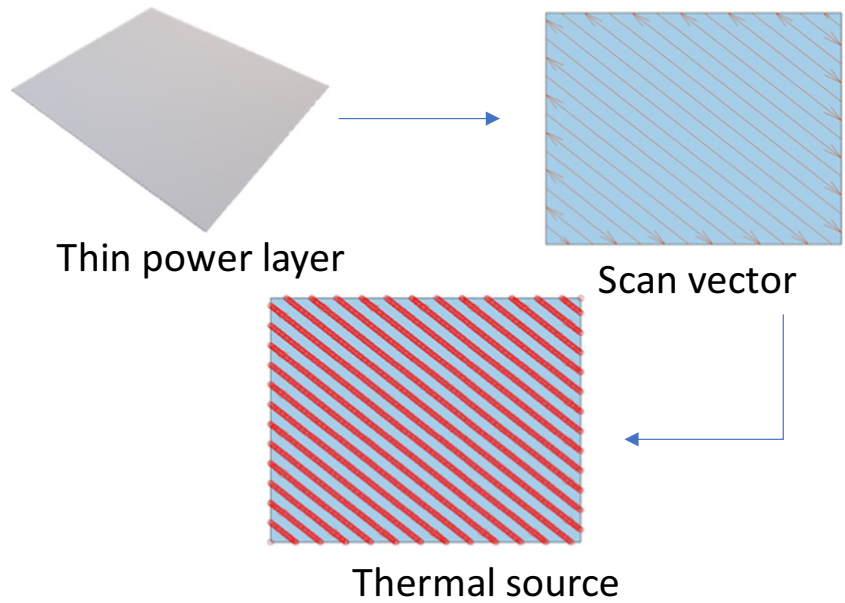
In our study, we consider two different layer geometries. The first is a rectangular grid indicated in Fig. 9. The second is shown in Fig. 10b.

For the airfoil case, additional constraints defining airfoil boundaries ( $A_j x + B_j y + C_j \leq 0 \quad \forall j = 1, \dots, N$ ) are included in the problem constraints. However, no meta-models are fitted on these boundary constraints since its functions are known and given.

Fig. 7 The thermal temperature of a discrete point changes with time. The horizontal line shows the melting temperature of Ti-6Al-4 V in °C



**Fig. 9** Standard scan pattern for a rectangular coupon



With these definitions, the additive process design problem can be written as

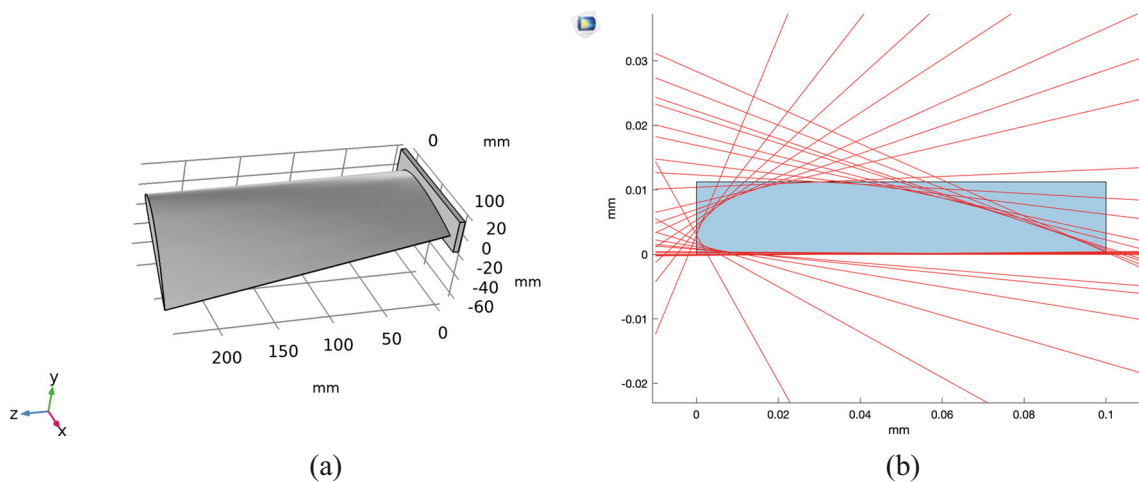
$$\begin{aligned}
 & \max_{p,v,h_s} v h_s \\
 & s.t. : \min_{x_1,x_2} n_m(p, v, h_s, x_1, x_2) \geq 3, \\
 & \min_{x_1,x_2} t_m(p, v, h_s, x_1, x_2) \geq 0.001 \\
 & \max_{x_1,x_2} t_m(p, v, h_s, x_1, x_2) \leq 0.02
 \end{aligned} \tag{20}$$

where  $p$  is the laser scanning power,  $v$  is the laser scanning velocity, and  $h_s$  is the hatch spacing of the scanning vector. Therefore, the objective relates to the square of the overall completion rate.

For both cases, the objectives are maximization of  $v h_s$ , which can be defined as the laser printing “efficiency”. Similarly, we adapted the other methods where a grid of 16 or 31 environmental variable points was applied so

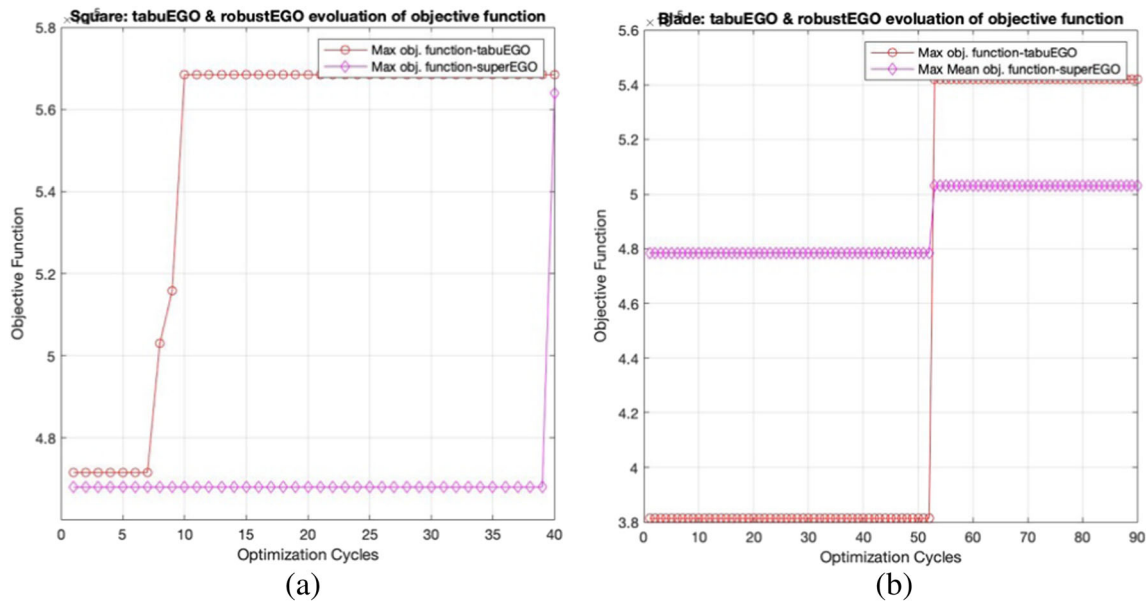
that the robust problem was approximated as a simple constrained problem with constraints for each set of environmental conditions. The results from the rectangular and airfoil cases show that our method offers a promising process parameter combination while satisfying thermal quality constraints. The benefits of the second problem are quite significant.

Figure 11 and Table 6 show the results from a single replication (for time reasons) of the alternative method solution values plotted against iteration. Compared with robust EGO, tabu EGO produces a solution with almost double the velocity  $\times$  hatch spacing or the square root of two times the production rate (41% faster). Such benefits could make a significant difference in making AM viable for practical problems. Note that we are plotting the proven best solution and not the predicted



**Fig. 10** a 3D airfoil and b 2D cross section of N-10 with constraints defining the boundary





**Fig. 11**  $f_{best}$  vs. iteration of tabu EGO including all alternative methods for 2 application cases. Alternative methods include tabu EGO and superEGO. **a** and **b** The comparison plots for applications 1 and 2, respectively. In these problems, the objective is maximization

best, which makes our plot relatively constant (e.g., compared with Ferreira and Serpa 2018).

### 6 Discussion

We propose the incorporation of tabu list penalties in infill optimization to improve the computational performance of EGO methods for a variety of black-box optimization problems. This approach decreases the possibility that experimental effort is directed at or near previous points in the decision or environmental variable spaces, with an additional sub-step to identify and store the currently relevant conditions. This frees up evaluations to search new points resulting in faster convergence and improved solution quality. To comprehensively evaluate the proposed methods, we compare the performance of the proposed tabu EGO with alternatives, including DIRECT, superEGO, robust EGO, fmincon, TPE and for unconstrained, constrained, and robust constrained problems, selecting the appropriate method alternative for each type of problem.

The results show the generally better performance of tabu EGO in terms of convergence speed and solution quality, given a fixed run size, for the majority of our test problems. For all the

tests for which tabu EGO did not perform the best, one other solution method (TPE, fmincon) provided better results. Our applications to thermal-based optimization of additive manufacturing suggest that the benefits from tabu EGO could be quite relevant practical. Tabu EGO produces solutions in one case that produces (almost) 41% improved speed of part production compared with robust EGO. Note that the objective function used is proportional to the square of the speed.

Note that one of the features of the original EGO algorithm is its ability to locate all the minima when there are multiple minima. One concern for the tabu EGO algorithm is that the penalty term will prohibit the algorithm from sampling near these minimums. Thus, it might never discover there are multiple minima. However, with tabu EGO, it is possible to move within the current epsilon value of another point. Such moves are merely penalized. Both the penalty and  $\epsilon$  are dynamic. This explains why our new approach often results in an ongoing reduction in the tabu list  $\epsilon$  value. More importantly, tabu EGO converges for all instances and all test functions except for test function #4 (G06). Our performance is superior to all the alternatives that we consider except on test function #4. Note that superEGO repeatedly fails to converge on test function #2 (Table 7). This occurs presumably because the failure

**Table 6** Case study-based comparison of solutions after 100 evaluations for application 1 and 150 evaluations for application 2 (the objectives are maximization)

Application	Method	Best solution $x_d$	$f_{best}$
1	robust EGO	288.519,1.833,3.080e-5,0.004,0.003	5.640e-5
	tabu EGO	258.889,1.877,3.030e-5,0.005,0.007	<b>5.690e-5</b>
2	robust EGO	229.259,1.744,2.880e-5,0.069,6.710e-3	5.030e-5
	tabu EGO	261.852,1.789,3.030e-5,0.034,0.007	<b>5.420e-5</b>

**Table 7** Comparison of solutions of tabu EGO and superEGO for Griewank and G08 functions over 10 different DoEs

Test functions	Griewank		G08	
	tabu EGO	superEGO	tabu EGO	superEGO
DoE				
1	0.0000	0.4705	-0.0841	-0.0938
2	0.3595	0.0000	-0.0848	-0.0848
3	0.0000	0.2664	-0.0958	-0.0958
4	0.0000	0.9444	-0.0955	-0.0956
5	0.2713	0.0000	-0.0646	-0.0787
6	0.2751	0.1621	-0.0501	-0.0261
7	0.0537	0.3535	N/F	N/F
8	0.1427	0.4939	-0.0267	-0.0267
9	0.2812	0.155	N/F	N/F
10	0.0000	0.0285	-0.0933	-0.0940
Mean	0.1384	0.2874	-0.0744	-0.0744
STD	0.1451	0.2940	0.0251	0.0302
Best	0.0000	0.0000	-0.0958	-0.0958

of the Kriging model based on empirical estimation of the smoothness parameter can cause convergence issues that are mitigated through a more widespread search.

There are several opportunities for future research. First, tabu EGO can be extended to address noisy optimization problems (e.g., Huang et al. 2006a) and multi-objective optimization problems (e.g., Li et al. 2008 or for variance or quality objectives and exploration Allen et al. 2018). Also, both single objective and multi-objective problems could employ techniques such as parallel computing for speed and efficiency improvement. Second, multiple fidelity optimization (see Huang and Allen 2005; Huang et al. 2006b) and problems involving two or more types of simulation can be studied, e.g., thermal and finite element method (FEM) simulations of additive manufacturing. Third, more elements of the standard tabu search structure can be explicitly incorporated to extend tabu EGO. These elements include short and long-term tabu lists, expiration conditions for points on the tabu list, candidate lists, and filter strategies. We have argued that the EI function approaches are likely to result in better performance than the local search strategies common in tabu searches. However, this can be studied empirically. Fourth, the additive manufacturing application can be explored more thoroughly, including experimental data and addressing layer interactions and part structure ending conditions.

**Funding** The funding source was the National Science Foundation Grant #1912166.

### Compliance with ethical standards

**Conflict of interest** We authors have no interests relating to this publication and no conflicts of interests.

**Replication of results** Throughout this paper, we have included pseudo-codes of key tabu EGO method and implementation instruction on numerical examples. These pseudo-codes are useful in the replication of our results. The interested reader should also ask Dr. Sasena or Dr. Regis directly if he/she is willing to use the code of superEGO or ConstrLMSRBF. Also, here is the download link for hyperopt: <https://github.com/hyperopt/hyperopt>.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### References

- Allen TT, Sui Z, Akbari K (2018) Exploratory text data analysis for quality hypothesis generation. *Qual Eng* 30(4):701–712
- Audet C, Denni J, Moore D, Booker A, Frank P (2000) A surrogate-model-based method for constrained optimization. In: 8th symposium on multidisciplinary analysis and optimization 4891
- Ben-Tal A, El Ghaoui L, Nemirovski A (2009) *Robust optimization*. Princeton University Press, Princeton
- Bergstra J, Yamins D, and Cox D (2013) Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pp. 115–123
- Bertsimas D, Nohadani O, Teo KM (2010) Non-convex robust optimization for problems with constraints. *INFORMS J Comput* 22(1): 44–58. <https://doi.org/10.1287/ijoc.1090.0319>
- Chelouah R, Patrick S (2000) Tabu search applied to global optimization. *EJOR* 123(2):256–270
- Duarte A, Marti A, Glover F, Gortazar F (2011) Hybrid scatter tabu search for unconstrained global optimization. *Ann Oper Res* 183(1):95–123
- Ferreira WG, Serpa AL (2018) Ensemble of metamodels: extensions of the least squares approach to efficient global optimization *Structural and Multidisciplinary Optimization* 57:131–159. <https://doi.org/10.1007/s00158-017-1745-x>
- Forrester AI, Keane AJ, Bressloff NW (2006) Design and analysis of “noisy” computer experiments[J]. *AIAA J* 44(10):2331–2339
- Frazier WE (2014) Metal additive manufacturing: a review. *J Mater Eng Perform* 23(6):1917–1928
- Glover F (1986) Future paths for integer programming and links to artificial intelligence. *Comput Oper Res* 13(5):533–549
- Glover F (1989) Tabu search—part 1. *ORSA J Comput* 1(2):190–206
- Glover F (1990) Tabu search—part 2. *ORSA J Comput* 2(1):4–32
- Huang D, Allen TT (2005) Design and analysis of variable fidelity experimentation applied to engine valve heat treatment process design. *J R Stat Soc: Series C (Applied Statistics)* 54(2):443–463
- Huang D, Allen TT, Notz WI, Zheng N (2006a) Global optimization of stochastic black-box systems via sequential Kriging meta-models. *J Glob Optim* 34(3):441–466
- Huang D, Allen TT, Notz WI, Miller RA (2006b) Sequential kriging optimization using multiple-fidelity evaluations. *Struct Multidiscip Optim* 32(5):369–382

- Jones DR (2009a) Direct global optimization algorithm. *Encycl Optim* 1(1):431–440
- Jones DR (2009b) Direct global optimization algorithm. *Encycl Optim* 725–35
- Jones D, Schonlau M, Welch W (1998) Efficient global optimization of expensive black-box functions. *J Glob Optim* 13: 455–492
- Kazemi M, Wang GG, Rahnamayan S, Gupta K (2010) Constraint importance mode pursuing sampling for continuous global optimization. In *ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference Jan 1* (pp. 325–334). Am Soc of Mech Eng Digital Collection
- Li M, Li G, Azarm S (2008) A kriging meta-model assisted multi-objective genetic algorithm for design optimization. *J Mech Des* 130(3):031401
- Lin G, Guan J, Li Z, Feng H (2019) A hybrid binary particle swarm optimization with tabu search for the set-union knapsack problem. *Expert Syst Appl* 135:201–211
- Liu H, Xu S, Chen X, Wang X, Ma Q (2017) Constrained global optimization via a DIRECT-type constraint-handling technique and an adaptive metamodeling strategy. *Struct Multidiscip Optim* 55(1): 155–177
- Ma H, Simon D (2017) *Evolutionary computation with biogeography-based optimization*. ISTE, Limited
- Matheron G (1963) Principles of geostatistics. *Econ Geol* 58:1246–1266
- Regis RG (2011) Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. *Comput Oper Res* 38(5):837–853
- Santner TJ, Williams BJ, Notz WI (2018) *The design and analysis of computer experiments*. Springer, New York
- Sasena MJ (2002) Flexibility and efficiency enhancements for constrained global design optimization with Kriging approximations. Ph. D. dissertation, University of Michigan
- Sasena MJ, Papalambros P, Goovaerts P (2002) Exploration of metamodeling sampling criteria for constrained global optimization. *Eng Optim* 34(3):263–278
- Schwalbach EJ, Donegan SP, Chapman MG, Chaput KJ, Groeber MA (2019) A discrete source model of powder bed fusion additive manufacturing thermal history. *Addit Manuf* 25:485–498
- The Math Works, Inc. (2020). MATLAB (Version 2020a) [Computer software]. <https://www.mathworks.com/help/optim/ug/fmincon.html> Accessed 22 May 2020
- ur Rehman S, Langelaar M (2017) Expected improvement-based infill sampling for global robust optimization of constrained problems. *Opt Eng* 18(3):723–753
- Viana FA, Haftka RT, Watson LT (2013) Efficient global optimization algorithm assisted by multiple surrogate techniques *Journal of Global Optimization* 56:669–689
- Wang L, Shan S, Wang GG (2004) Mode-pursuing sampling method for global optimization on expensive black-box functions. *Eng Optim* 36(4):419–438

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.