A multi-fidelity RBF surrogate-based optimization framework for computationally expensive multi-modal problems with application to capacity planning of manufacturing systems

Jin Yi¹ · Yichi Shen² · Christine A. Shoemaker³

Received: 9 November 2019 / Revised: 14 February 2020 / Accepted: 12 March 2020 / Published online: 17 May 2020
© The Author(s) 2020

Abstract

This paper presents a multi-fidelity RBF (radial basis function) surrogate-based optimization framework (MRSO) for computationally expensive multi-modal optimization problems when multi-fidelity (high-fidelity (HF) and low-fidelity (LF)) models are available. The HF model is expensive and accurate while the LF model is cheaper to compute but less accurate. To exploit the correlation between the LF and HF models and improve algorithm efficiency, in MRSO, we first apply the DYCORDS (dynamic coordinate search algorithm using response surface) algorithm to search on the LF model and then employ a potential area detection procedure to identify the promising points from the LF model. The promising points serve as the initial start points when we further search for the optimal solution based on the HF model. The performance of MRSO is compared with 6 other surrogate-based optimization methods (4 are using a single-fidelity surrogate and the rest 2 are using multi-fidelity surrogates). The comparisons are conducted on a multi-fidelity optimization test suite containing 10 problems with 10 and 30 dimensions. Besides the benchmark functions, we also apply the proposed algorithm to a practical and computationally expensive capacity planning problem in manufacturing systems which involves discrete event simulations. The experimental results demonstrate that MRSO outperforms all the compared methods.

Keywords Surrogate-assisted optimization · Multi-fidelity surrogate · Radial basis function · Computationally expensive problems

1 Introduction

Computationally expensive multi-modal black-box optimization problems arise in many science and engineering areas. For example, in the field of complex products design

optimization, the optimization analysis is driven by expensive computer simulation codes (e.g., finite element analysis (FEA) and computational fluid dynamics (CFD)), which are used to simulate the physical processes and to evaluate the design solutions (Simpson et al. 2004). Another example can be found in the hyper-parameter tuning process of machine learning algorithms. Each evaluation of the hyper-parameter configuration needs to run the whole training and testing process, which may take minutes or hours (Ilievski et al. 2017). A computationally expensive multi-modal black-box optimization problem can be formulated as follows:

$$\begin{aligned} &\text{minimize: } f(\mathbf{x}) \\ &\text{subject to: } L_i \leq x_i \leq U_i, 1 \leq i \leq d \end{aligned} \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_d)$ is the decision vector, d is the number of decision variables, and x_i ($1 \leq i \leq d$) is the i th decision variable. L_i and U_i are the lower

Responsible Editor: Nathalie Bartoli

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s00158-020-02575-7>) contains supplementary material, which is available to authorized users.

✉ Christine A. Shoemaker
shoemaker@nus.edu.sg

¹ Environmental Research Institute, National University of Singapore, Singapore, Singapore

² Institute of Operation Research and Analytics, National University of Singapore, Singapore, Singapore

³ Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore, Singapore

and upper bounds on x_i . Computationally expensive multi-modal black-box optimization problems have the following challenges: (1) Every single evaluation of the objective function is expensive, which may take minutes, hours, or even days. Thus, in these cases only a relatively small number of evaluations are possible; (2) no derivative information can be obtained from the problem. Therefore, the gradient-based optimization methods cannot be directly applied. (3) The problem may have multiple local minima.

In recent years, surrogate-assisted global optimization (SAGO) methods have become very popular for computationally expensive black-box optimization problems. The basic idea of SAGO is to build a cheap approximate model (called a surrogate model or metamodel) for the expensive objective function and use this cheap model to replace some of the expensive evaluations, in the optimization search this can dramatically reduce the computational cost. Depending on the ways of using the surrogates, the SAGO methods can be classified into two categories: the offline mode and the online mode. In the offline mode, a surrogate is built at the beginning of the optimization process and will not be updated. In contrast, the online model keeps updating the surrogate when new evaluations are available. The majority of SAGO methods fall into the second category. More details about SAGO methods can be found in the literature reviews (Jin 2011; Haftka et al. 2016).

The surrogates used in the aforementioned SAGO methods are built solely on the high-fidelity (HF) models. Here, the term “fidelity” means how accurate or how close the model is for describing the original objective function. If a model is accurate and highly trusted, then it is an HF model. Otherwise, it is a low-fidelity (LF) model. Although these SAGO methods have been proved to be effective in various applications (Li et al. 2017; Liu et al. 2017, 2018; Han et al. 2018; Ali et al. 2018; Redouane et al. 2019), their shortcomings appear when dealing with extremely expensive problems, or medium- to high-dimensional decision vectors. To ensure higher prediction accuracy of the surrogate model, the search space should be space-filling sampled by sufficient sample points. However, it is impractical for both extremely expensive and medium- to high-dimensional decision vectors due to the limitation of computational budgets and the “curse of dimensionality.” Therefore, there are still limitations for the SAGO to tackle these kinds of computationally expensive problems.

To fill this gap, the multi-fidelity optimization (MFO) methods have received much attention in recent years and have been applied to various fields (Liu et al. 2016; Kandasamy et al. 2016; Bonfiglio et al. 2018; Habib et al. 2019; Liu et al. 2019; Yi et al. 2019; Zhang et al. 2019a; Yong et al. 2019; Zhang et al. 2018; Ding and Kareem 2018; Tao and Sun 2019). Instead of only building a single-fidelity surrogate-based on the HF model, MFO

methods also utilize a LF model, which is less accurate but much cheaper and easier to compute. By using a large number of cheap LF evaluations and a small number of expensive HF evaluations, MFO can obtain better solutions with less computational cost. In addition, the LF models are easy to access. For example, the LF models of the airfoil shape optimization can be created by using coarse meshes (Zhou et al. 2017). There are two strategies among common MFO methods. In the first strategy, a variable-fidelity surrogate model is built to minimize the discrepancy between the LF model and the HF model. A lot of previous studies discussed how to build accurate variable-fidelity models (Zheng et al. 2015; Zhou et al. 2017; Durantin et al. 2017; Liu et al. 2018; Zheng 2018; Song et al. 2019). Zhang et al. (2018) proposed a variable-fidelity expected improvement criterion on a hierarchical Gaussian process model, which can adaptively select new samples on both LF and HF models. Yi et al. (2019) proposed a variable-fidelity RBF surrogate-assisted harmony search algorithm, where a multi-level screening mechanism based on non-dominated sorting is used to strictly control the numbers of LF and HF evaluations. In the second strategy, multiple surrogates are built for models of different fidelity levels and some recent research follows this line. Liu et al. (2016) developed a multi-fidelity surrogate model-assisted memetic differential evolution (DE) method, where the Gaussian process models are built based on both the LF and HF models. Kandasamy et al. (2016) formulated the MFO task as a multi-fidelity bandit problem and proposed a multi-fidelity Bayesian optimization method with the upper confident bound criterion.

We propose MRSO, a multi-fidelity RBF surrogate-based optimization framework that utilizes the original RBF surrogate-assisted dynamic coordinate search algorithm (DYCORS) (Regis and Shoemaker 2013), in which the surrogate is built on single-fidelity models. This study aims to further improve the performance and speed-up the optimization process for computationally expensive multi-modal black-box problems.

The proposed MFO framework has three main steps: In the first step, the original DYCORS algorithm is used to search on the LF model. Secondly, a potential area detection procedure is adopted to identify the potential points. Finally, the obtained potential points serve as the initial starting points for the DYCORS algorithm to further search on the HF model. The effectiveness of the proposed approach is validated by the comparison with 6 other surrogate-assisted global optimization methods include both single-fidelity and multi-fidelity methods. A capacity planning problem in manufacturing systems involving discrete event simulations is also used to demonstrate the superior performance of the proposed approach.

The main contributions of this study are as follows:

- (1) MRSO starts the search from the LF model and quickly identifies the potential search area. Then, it further searches on the HF model from the potential area. With the help of LF model, MRSO can further save the computational budget and speed-up the optimization process.
- (2) To link the search on LF and HF models, we innovatively develop a potential area detection procedure to identify the guesses for local optimal locations, which serve as the starting points for the HF run.
- (3) Empirically, we demonstrate that the proposed MRSO approach outperforms 6 other surrogate-assisted global optimization methods including both single-fidelity and multi-fidelity surrogate-based methods on a series of problems, including 10 numerical benchmark problems from the MFO test suite. Moreover, MRSO has been successfully applied to a complex capacity planning problem in manufacturing systems involving discrete event simulations.

The remainder of this paper is organized as follows. The prior research related to this research is given in Section 2. Section 3 introduces the proposed MRSO approach. Section 4 presents the numerical experiments. An application on the capacity planning problem is discussed in Section 5. Finally, Section 6 gives the concluding remarks.

2 Surrogate-assisted global optimization and DYCORS algorithm

2.1 Surrogate-assisted global optimization

Most of the surrogate-assisted global optimization methods have the same framework shown in Fig. 1. Note that all methods here are single-fidelity surrogate-based methods (with the surrogate built solely on HF model). At the start of the optimization, a space-filling experimental design procedure (e.g., Latin hyper-cube design (Kenny et al. 2000)) is carried out to generate initial samples over the whole domain, and these initial samples are evaluated by the expensive HF models. Once this step is finished, a surrogate model is built based on the collected initial data A (i.e., a set of input-output pairs A). In the next step, a user-defined acquisition function is used to decide the next evaluation point x_E based on the surrogate information. Once the next evaluation point is found, it will be further evaluated by the expensive HF model, and this new data will be added into the database A . When the stopping criterion is not satisfied, the algorithms will go back to step 2 to update the surrogate model and repeat step 2 to step 4 (Fig. 1); otherwise, the algorithms will terminate and output the current best solution. Usually, the maximum

number of HF evaluations is used as the stopping criterion since the surrogate-assisted global optimization methods are targeted for computationally expensive problems with a limited computational budget.

The key ingredients of surrogate-assisted optimization methods for expensive multi-modal functions are in step 2 and step 3 (Fig. 1), which are (a) surrogate model, (b) optimization algorithm used to search on the surrogate, and (c) its corresponding acquisition function. There are several different choices for each of the key ingredients. For the choice of the surrogate model, the most widely used models are Gaussian process model (or Kriging model) (Shahriari et al. 2015; Zhang et al. 2019b) and radial basis function model (RBF) (Regis and Shoemaker 2007, 2013; Regis 2014). Other methods that could also be used include the multivariate adaptive regression splines (MARS) (Friedman et al. 1991), support vector regression (SVM) (Smola and Schölkopf 2004), artificial neural networks (ANN) (Villarrubia et al. 2018), and random forest (RF) (Wang and Jin 2018). There are papers showing that MARS regression is not efficient as an RBF surrogate on the specific tested problems (Müller and Shoemaker 2014). For the optimization algorithm used to search on the surrogate surface, the popular methods are under three big categories: (a) global optimization methods, (b) local optimization methods, and (c) random sampling. For global optimization methods, a genetic algorithm (GA) (Whitley 1994) or a particle swarm optimization (PSO) (Kennedy 2010) can be used. Since the $f(x)$ is multi-modal, the surrogate can also be multi-modal so the second approach is to use local optimization methods and combine with a multi-start strategy. The local optimization methods can include Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Battiti and Masulli 1990), Nelder-Mead simplex algorithm, and Newton conjugate gradient algorithm. In addition to these two methods, some previous researches applied random sampling to generate a fixed number of random candidates that are sorted on basis of the value of acquisition function, and the best candidate is selected for the next evaluation (Regis and Shoemaker 2007, 2013; Hutter et al. 2009). For the acquisition function, some popular options are expected improvement (EI) (Jones et al. 1998), probability of improvement (POI) (Snoek et al. 2012), upper bound of confidence (UCB), entropy search (ES) (Hennig and Schuler 2012), and the weighted score criterion (Regis and Shoemaker 2013), which will be introduced in the next subsection).

2.2 The DYCORS algorithm

The DYCORS algorithm proposed by Regis and Shoemaker (2013) is an efficient surrogate-assisted global optimization algorithm for computationally expensive problems.

Algorithm 1 gives the pseudo-code of DYCORS algorithm. DYCORS has three main features: (1) it adopts the RBF model as the surrogate, (2) it generates the candidate points by perturbing the current best solution with a predefined probability, and (3) it uses the weighted score of the surrogate prediction (i.e., exploitation) and the minimum distance to previous evaluated points (i.e., exploration) as the acquisition function. Details are given in the following sections.

Algorithm 1 DYCORS algorithm (Regis and Shoemaker 2013).

- 1: **Input:** The expensive black-box problem $f = f(\mathbf{x})$ defined in domain of $D = \{\mathbf{x} | L_i \leq x_i \leq U_i, 1 \leq i \leq d\} \subseteq \mathbb{R}^d$; the maximum number of expensive evaluations is set to max_n ; surrogate model is set to RBF; Number of candidate points in each iteration N_{can} ; Initial step size is set to δ_{init} , maximum and minimum step size are set to δ_{max} and δ_{min} ; the tolerance for the number of consecutive failed and consecutive succeed iterations $\mathcal{T}_f$ and $\mathcal{T}_s$.
- 2: **Output:** The found best solution $\mathbf{x}_{best}$ so far.
- 3: Set $n = 0$
- 4: Generate n_0 initial points using space-filling experiment design and evaluate them by expensive function f . Identify the best solution $\mathbf{x}_{best}$ so far and let $f_{best} = f(\mathbf{x}_{best})$. Let $\mathcal{A}_0 = \{(\mathbf{x}_1, f(\mathbf{x}_1)), (\mathbf{x}_2, f(\mathbf{x}_2)), \dots, (\mathbf{x}_{n_0}, f(\mathbf{x}_{n_0}))\}$, set $\mathcal{A} = \mathcal{A}_0$, $n = n + n_0$, $\delta = \delta_{init}$, $C_s = 0$ and $C_f = 0$.
- 5: **while** $n \leq max_n$ **do**
- 6: Build/Update the RBF surrogate $s_n(\mathbf{x})$ using the collected data $\mathcal{A}$.
- 7: Set/Update the probability of selecting the coordinate to perturb $p_n = \phi(n)$.
- 8: **for** $i = 1$ to N_{can} **do**
- 9: **for** $j = 1$ to d **do**
- 10: **if** $rand < p_n$ **then**
- 11: $x_{can}[i, j] = x_{best}[j] + z(\delta)$ /*perturbing around the current best solution*/
- 12: **end if**
- 13: **end for**
- 14: **end for**
- 15: Select the next evaluation point $\mathbf{x}_{n+1}$ from $\mathbf{x}_{can}$ using the **next evaluation point selection procedure** (see Algorithm 2).
- 16: Evaluate the selected point $f(\mathbf{x}_{n+1})$
- 17: **if** $f(\mathbf{x}_{n+1}) < f_{best}$ **then**
- 18: $f_{best} = f(\mathbf{x}_{n+1})$, $\mathbf{x}_{best} = \mathbf{x}_{n+1}$. /*Update current best solution*/
- 19: $C_s = C_s + 1$, $C_f = 0$.
- 20: **else**
- 21: $C_s = 0$, $C_f = C_f + 1$.
- 22: **end if**
- 23: **if** $C_s > \mathcal{T}_s$ **then**
- 24: $\delta = \min(2 \cdot \delta, \delta_{max})$, and $C_s = 0$
- 25: **end if**
- 26: **if** $C_f > \mathcal{T}_f$ **then**
- 27: $\delta = \delta/2$, and $C_f = 0$
- 28: **end if**
- 29: Update information. $\mathcal{A} = \mathcal{A} \cup \{\mathbf{x}_{n+1}\}$, $n = n + 1$
- 30: **end while**

Fig. 1 Basic flowchart of surrogate-assisted global optimization

2.2.1 The RBF model

The RBF model serves as the surrogate in DYCORS algorithm. The RBF is a real-valued function whose value depends only on the distance between the input point to some fixed given point.

Given a set of n sampling points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^d$ in d -dimensional space, and the vector $(f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_n))$ are the corresponding objective function values, then the RBF approximation can be obtained by the following:

$$s(\mathbf{x}) = \sum_{i=1}^n \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}) \quad (2)$$

where $\|\cdot\|$ is the Euclidean norm, $\lambda_i \in \mathbb{R}^d$, $p(\mathbf{x})$ is the linear polynomial, and $\phi(\cdot)$ is the basis function. $\phi(\cdot)$ has several different forms, and in this study, we use the cubic form $(\phi(r) = r^3)$ since previous study suggested that cubic RBFs might be more suitable than other forms of RBFs for surrogate-based optimization (Wild and Shoemaker 2013).

To determine the parameters (λ and c) of the RBF model in (2), we define the matrix $\Phi \in \mathbb{R}^{n \times n}$: $\Phi_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$, $i, j = 1, 2, \dots, n$, and matrix $P \in \mathbb{R}^{n \times (d+1)}$. The vector in the i th row of P is $[1, \mathbf{x}_i]$. Then, λ and c can be obtained by solving the following equation (Powell M 1992):

$$\begin{bmatrix} \Phi & P \\ P^T & \mathbf{0}_{(d+1) \times (d+1)} \end{bmatrix} \begin{bmatrix} \lambda \\ c \end{bmatrix} = \begin{bmatrix} F \\ \mathbf{0}_{(d+1) \times 1} \end{bmatrix} \quad (3)$$

where $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n) \in \mathbb{R}^n$, $F = (f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_n))^T$, $\mathbf{0}_{(d+1) \times (d+1)} \in \mathbb{R}^{(d+1) \times (d+1)}$ is a zero matrix, $c = (c_1, c_2, \dots, c_{(d+1)}) \in \mathbb{R}^{(d+1)}$ is

Fig. 2 Flowchart of the proposed MRSO approach using LF and HF models

the coefficient vector of the linear polynomial $p(x)$, $\mathbf{0}_{(d+1)} \in \mathbb{R}^{(d+1)}$ is a zero vector.

2.2.2 Candidate points generating

Lines 8 to 14 outlined in Algorithm 1 give the procedure of generating candidate points. The basic idea is to randomly perturb on some or all the coordinates of the best solution found so far ($\mathbf{x}_{best}$). The perturbing process is controlled by a probability $p_n = \phi(n)$. $\phi(n)$ is defined by the following equation:

$$\phi(n) = \phi_0 \cdot \frac{1 - \ln(n - n_0 + 1)}{\ln(\max_n - n_0)}, \quad n_0 \leq n \leq \max_n - 1. \quad (4)$$

where $\phi_0 = \min(20/d, 1)$. It can be found from (4) that p_n will be smaller when n is larger, which means DYCORDS perturbs most of the coordinates at the beginning but perturbs less dimensions at the end of the optimization process. Line 11 in Algorithm 1 is the perturbation operation, where $z(\delta)$ is a randomly generated number by a normal distribution $\mathcal{N}(0, \delta^2)$.

2.2.3 Selection of the next evaluation point

Line 15 in Algorithm 1 is the procedure for selecting the next evaluation point. The working steps are shown in Algorithm 2. In this procedure, the next evaluation point is selected according to the weighted score of the RBF prediction and the minimum distance to previously evaluated points. Based on line 13 of Algorithm 2, a larger value of ω encourages exploitation while a smaller value

of ω encourages exploration. To maintain the balance between exploration and exploitation, the ω in DYCORDS is cycling through a set of weights, which is denoted as $\gamma = \{\gamma_1, \dots, \gamma_s\}$, where $0 \leq \gamma_1 \leq \dots \leq \gamma_s \leq 1$. Note that this procedure is computationally efficient and its computational complexity is only $O(n \cdot N_{can})$.

Algorithm 2 Next evaluation point selection procedure (Regis and Shoemaker 2013).

- 1: **for** $i = 1$ to N_{can} **do**
- 2: Compute the RBF prediction of $\mathbf{x}_{can}[i]$ as $s_n(\mathbf{x}_{can}[i])$
- 3: **for** $k = 1$ to n **do**
- 4: Compute the Euclidean distance between $\mathbf{x}_{can}[i]$ and $\mathbf{x}_k$, denoted as $d(\mathbf{x}_{can}[i], \mathbf{x}_k)$.
- 5: **end for**
- 6: Find the minimum distance between $\mathbf{x}_{can}[i]$ and previously evaluated points, $\Delta_n(\mathbf{x}_{can}[i]) = \min_{1 \leq k \leq n} \{d(\mathbf{x}_{can}[i], \mathbf{x}_k)\}$
- 7: **end for**
- 8: $s_n^{max} = \max_{1 \leq i \leq N_{can}} \{s_n(\mathbf{x}_{can}[i])\}$, $s_n^{min} = \min_{1 \leq i \leq N_{can}} \{s_n(\mathbf{x}_{can}[i])\}$
- 9: $\Delta_n^{max} = \max_{1 \leq i \leq N_{can}} \{\Delta_n(\mathbf{x}_{can}[i])\}$, $\Delta_n^{min} = \min_{1 \leq i \leq N_{can}} \{\Delta_n(\mathbf{x}_{can}[i])\}$
- 10: **for** $i = 1$ to N_{can} **do**
- 11: $V_n^R(\mathbf{x}_{can}[i]) = (s_n(\mathbf{x}_{can}[i]) - s_n^{min}) / (s_n^{max} - s_n^{min})$, if $s_n^{max} \neq s_n^{min}$, and $V_n^R(\mathbf{x}_{can}[i]) = 1$ otherwise. /*compute score for the prediction criterion*/
- 12: $V_n^D(\mathbf{x}_{can}[i]) = (\Delta_n^{max} - \Delta_n(\mathbf{x}_{can}[i])) / (\Delta_n^{max} - \Delta_n^{min})$, if $\Delta_n^{max} \neq \Delta_n^{min}$, and $V_n^D(\mathbf{x}_{can}[i]) = 1$ otherwise. /*compute score for the distance criterion*/
- 13: $w_n(\mathbf{x}_{can}[i]) = \omega \cdot V_n^R(\mathbf{x}_{can}[i]) + (1 - \omega) \cdot V_n^D(\mathbf{x}_{can}[i])$. /*compute the weighted score*/
- 14: **end for**
- 15: Find $\mathbf{x}_{n+1} = \argmin_{1 \leq i \leq N_{can}} \{w_n(\mathbf{x}_{can}[i])\}$

3 Proposed MRSO algorithm

In this section, we introduce an efficient MRSO framework that uses some features from a single-fidelity surrogate-based global optimization algorithm but can further reduce the number of expensive objective function evaluations required for accurate solutions, beyond what is achievable with single-fidelity methods. First, the main steps of the proposed MRSO are given, and then some important features of MRSO are introduced in detail. For the convenience of explaining the algorithm, we define several notations, as shown in Table 8 from the Appendix.

3.1 The framework of MRSO

The flowchart of MRSO is depicted in Fig. 2 and the main steps are as follows:

- Step 1: Space-filling experiment design for the LF model. Generate a sampling set $\mathcal{D}_L = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_0}\}$ using a Latin hyper-cube design for the LF model.
- Step 2: Run LF model to obtain the LF response values $f_L(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathcal{D}_L, i = 1, 2, \dots, n_L$. Set LF model evaluation number n_L as $n_L = n_0$. Initialize the database Θ_L , set $\Theta_L = \{(\mathbf{x}_i, f_L(\mathbf{x}_i)), i = 1, 2, \dots, n_L\}$. Let $\mathbf{x}_L^*$ be the best solution found so far on the LF model and $f_{\text{best}}^L = f^L(\mathbf{x}_L^*)$.
- Step 3: While the termination condition is not satisfied. (e.g., $n_L < \max n_L$) do
- Step 3.1: Build (or update) the RBF surrogate $S_L(\mathbf{x})$ for the LF model using database Θ_L .
- Step 3.2: Randomly generate N_{can} candidate points denote as $C_L = \{\mathbf{x}_1^L, \mathbf{x}_2^L, \dots, \mathbf{x}_{N_{\text{can}}}^L\}$ around the current best solution $\mathbf{x}_L^*$ on the LF model by perturbing on part of the dimensions.
- Step 3.3: Select the next evaluation point $\mathbf{x}_{\text{nl}}$ from C_L by the weighted score criterion. The weight cycling pattern is denoted as γ .
- Step 3.4: Evaluate $\mathbf{x}_{\text{nl}}$ using the LF model and update the current best solution. If $f^L(\mathbf{x}_{\text{nl}}) < f_{\text{best}}^L$, then $\mathbf{x}_L^* = \mathbf{x}_{\text{nl}}$ and $f_{\text{best}}^L = f^L(\mathbf{x}_{\text{nl}})$.
- Step 3.5: Update Information. Set $\Theta_L = \Theta_L \cup \{\mathbf{x}_{\text{nl}}\}$ and set $n_L := n_L + 1$.
- Step 4: Potential area detection procedure.
- Step 4.1: Run an optimizer on surrogate $S_L(\mathbf{x})$ (i.e., run BFGS or GA for multiple trials, or run a multi-modal estimation of distribution algorithm). Store the final solutions into set $\mathcal{A}$, let $\mathcal{A} = \{(\mathbf{x}_i, S_L(\mathbf{x}_i)), i = 1, 2, \dots, m\}$, m is the number of solutions.
- Step 4.2: Use a cluster algorithm to classify the points from $\mathcal{A}$ into N_g groups. Identify the point with the best

surrogate prediction value in each group and they are denoted as $\{\mathbf{x}_k, k = 1, 2, \dots, N_g\}$.

- Step 4.3: Re-evaluate the obtained points in step 4.2 with the actual LF function and keep the top $Q\%$ points with best LF objective function values. Then, store these points and the best solution from the LF run into a set $\mathcal{D}_H$, let $\mathcal{D}_H = \{\mathbf{x}_L^*\} \cup \{\mathbf{x}_k, k = 1, 2, \dots, \lfloor N_g \cdot Q\% \rfloor\}$. Set $n_L := n_L + N_g$

- Step 5: Run HF model to obtain the HF response values $f_H(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathcal{D}_H, i = 1, 2, \dots, 1 + \lfloor N_g \cdot Q\% \rfloor$. Set HF model evaluation number n_H as $n_H = N_g$. Initialize the database Θ_H , set $\Theta_H = \{(\mathbf{x}_i, f_H(\mathbf{x}_i)), i = 1, 2, \dots, n_H\}$. Let $\mathbf{x}_H^*$ be the best solution on the HF model and $f_{\text{best}}^H = f_{\text{best}}^H(\mathbf{x}_H^*)$.
- Step 6: While the termination condition is not satisfied. (e.g., $n_H < \max n_H - n_L/R$, R is the computational cost ratio between HF and LF models) do

- Step 6.1: Build (or update) the RBF surrogate $S_H(\mathbf{x})$ for the HF model using database Θ_H .
- Step 6.2: Randomly generate N_{can} candidate points denote as $C_H = \{\mathbf{x}_1^H, \mathbf{x}_2^H, \dots, \mathbf{x}_{N_{\text{can}}}^H\}$ around the current best solution $\mathbf{x}_H^*$ on the HF model by perturbing on part of the dimensions.
- Step 6.3: Select the next evaluation point $\mathbf{x}_{\text{nh}}$ from C_H by the weighted score criterion. The weight cycling pattern is denoted as γ .
- Step 6.4: Evaluate $\mathbf{x}_{\text{nh}}$ using the HF model and update the current best solution. If $f^H(\mathbf{x}_{\text{nh}}) < f_{\text{best}}^H$, then $\mathbf{x}_H^* = \mathbf{x}_{\text{nh}}$ and $f_{\text{best}}^H = f^H(\mathbf{x}_{\text{nh}})$.
- Step 6.5: Update Information. Set $\Theta_H = \Theta_H \cup \{\mathbf{x}_{\text{nh}}\}$ and set $n_H := n_H + 1$.

- Step 7: Return the best solution $\mathbf{x}_H^*$ found so far.

3.2 Potential area detection procedure

A potential area detection procedure is proposed in MRSO. Its eventual aim is to identify multiple local optima, which might be in different area on the LF model and use them as the initial samples to drive the HF run. The motivation behind is that the LF and HF share some similarities; therefore, a better guess of the global optimum location using LF model could accelerate the optimization process on the HF model significantly.

The potential area detection procedure has three main steps. In the first step (Fig. 2, part II, step 4.1), an optimizer is used to search on the RBF surrogate $S_L(\mathbf{x})$ and identify the multiple local optimal solutions. In this study, three different methods have been utilized to complete this task. The first choice is to run a gradient-based algorithm (i.e., BFGS (Zhu et al. 1997)) for multiple trials from different

start points. The second choice is to run a metaheuristic method (i.e., GA) for multiple trials. The third option is to run a multi-modal optimization method (i.e., multi-modal estimation of distribution algorithm (Yang et al. 2016)) for a single trial. In the second step (Fig. 2, part II, step 4.2), a cluster algorithm is used to classify the final solutions obtained in step one into several different groups. Here, the number of groups is the same as the number of local optima, and hence it is problem-dependent. In this study, the DBSCAN (density-based spatial clustering of applications with noise) algorithm (Schubert et al. 2017) is selected as the clustering algorithm since it does not require specifying the number of groups in advance of the clustering. In the third step (Fig. 2, part II, step 4.3), the best point in each group is re-evaluated by the LF model and the top $Q\%$ (Q is set to 50 in this study) points are selected. This step is necessary because the LF surrogate is less trusted and it helps filter those points that have nice LF surrogate prediction but bad actual LF model values.

To better demonstrate the potential area detection procedure, we take the 2-D Styblinski-Tang function as example. In this case, the aim is to maximize the function objective, and the HF and LF models are defined as follows:

$$f_H(\mathbf{x}) = -\frac{1}{2} \sum_{i=1}^d (x_i^4 - 16x_i^2 + 5x_i). \quad (5)$$

Fig. 3 An illustration of the potential area detection process on 2-D Styblinski-Tang function. **a** Shows the landscape of LF model; **b** shows the landscape of HF model; **c** shows some points near multi-peaks are obtained by running GA to find maximum of the LF surrogate over multiple trials; **d** shows that all the peaks on the LF model are identified by the clustering algorithm

(a) Landscape of the LF model

(a) Landscape of the HF model

(c) Points obtained by multiple trials of GA on the LF model

(d) Multi-peaks are identified by cluster algorithm on the LF model

$$f_L(\mathbf{x}) = f_h(\mathbf{x}) + f_e(\mathbf{x}). \quad (6)$$

where $f_e(\mathbf{x}) = \eta \cdot f_h(\mathbf{x})$ is the error function and η is a uniform distributed random number between -0.1 and 0.1 .

Figure 3 shows the potential area detection process. Figure 3 a and b show the landscapes of the LF and HF models, respectively. It can be seen that the HF and LF models have similar landscapes. However, the HF model is smoother while the LF model is rougher. In the first step, a multi-modal metaheuristic/evolutionary algorithm is used to search on the RBF surrogate $S_L(\mathbf{x})$, and the final solutions are shown in Fig. 3c. Then, the DBSCAN algorithm is applied to divide these points into different groups, and the best points from different groups, which is shown in Fig. 3d, are selected as the initial samples for the HF run.

3.3 Restart strategy

The restart strategy is applied to our proposed MRSO. Based on the previous study, the restart strategy is simple but efficient (Regis and Shoemaker 2007). In MRSO, the restart operation is triggered if the step size δ in Algorithm 1 is smaller than a threshold value $\delta_{\min}$. Once the restart strategy is triggered, all previously evaluated points will be abandoned and the algorithm starts over the whole optimization process of DYCORDS from the experiment design.

4 Experimental results

In this section, the performance of proposed MRSO is validated on 10 benchmark functions with 10 and 30 dimensions and compared with 6 other surrogate-assisted optimization methods that include both single-fidelity and multi-fidelity methods.

4.1 Test problems

The benchmark functions adopted in this study are from the MFO test suite (Wang et al. 2017). In the test suite, a parameter α ranging from 0 to 10000 is introduced to describe the level of fidelity, where 0 represents the lowest level and 10000 represents the highest level. If the exact function is $f(\mathbf{x})$, then the approximation of $f(\mathbf{x})$ at fidelity level α can be formulated as follows:

$$\hat{f}(\mathbf{x}, \alpha) = f(\mathbf{x}) + f_e(\mathbf{x}, \alpha) \quad (7)$$

where $f_e(\mathbf{x}, \alpha)$ is the error function. The value of $f_e(\mathbf{x}, \alpha)$ is smaller if fidelity level α is higher. For example, the error will be zero if $\alpha = 10000$. In this study, we set $\alpha = 1000$ for the LF model, and $\alpha = 10000$ for the HF model. Three different types of error sources are introduced in the MFO test suite, including resolution errors, stochastic errors, and instability errors. The original test suite has 13 problems, but some of the problems have exactly the same expressions of HF and LF models. So, we choose the unique 10 problems as the test problems, which include P_1 – P_3 and P_7 – P_{13} from the MFO test suite. The detailed information of the test problems is given in Table 1.

4.2 Methods in comparison and experimental setup

The proposed MRSO is compared with 6 well-known surrogate-assisted optimization methods under two categories: single-fidelity methods and multi-fidelity methods.

Table 1 Information about the test problems

Function	Description	Range	Optimum
F ₁	P ₁ in MFO test suite	$[-1, 1]^d$	0
F ₂	P ₂ in MFO test suite	$[-1, 1]^d$	0
F ₃	P ₃ in MFO test suite	$[-1, 1]^d$	0
F ₄	P ₇ in MFO test suite	$[-1, 1]^d$	0
F ₅	P ₈ in MFO test suite	$[-1, 1]^d$	0
F ₆	P ₉ in MFO test suite	$[-1, 1]^d$	0
F ₇	P ₁₀ in MFO test suite	$[-1, 1]^d$	0
F ₈	P ₁₁ in MFO test suite	$[-1, 1]^d$	0
F ₉	P ₁₂ in MFO test suite	$[-1, 1]^d$	0
F ₁₀	P ₁₃ in MFO test suite	$[-1, 1]^d$	0

The single-fidelity methods include the DYCORS algorithm (Regis and Shoemaker 2013), the surrogate-assisted hierarchical particle swarm optimization (SHPSO) (Yu et al. 2018), the hybrid surrogate-based optimization using space reduction (HSOSR) (Dong et al. 2018), and the surrogate-guided differential evolution algorithm (S-JADE) (Cai et al. 2019). The multi-fidelity methods include the multi-fidelity Gaussian process and radial basis function-model-assisted memetic differential evolution (MGPMDE) (Liu et al. 2016), and the multi-fidelity Bayesian optimization (MF-GP-UCB) (Kandasamy et al. 2016). The MRSO and DYCORS are implemented based on the pySOT framework,¹ which is written in Python (Eriksson et al. 2019). The other methods are implemented in MATLAB. The code of MF-GP-UCB can be found from the GitHub website² and the code of S-JADE is provided by the original authors (Cai et al. 2019). Other algorithms are re-implemented by the authors of this paper according to the descriptions in the literature. The experiments are run on the high-performance computing platform at National University of Singapore.

Table 2 presents the parameter settings of the compared methods following the recommendations from the original literature. For a fair comparison, here we assume that the computational time of 10 LF evaluations is equal to one HF evaluation, as is common in industry (Liu et al. 2016; Koziel et al. 2014). Note that MRSO, MGPMDE, and MF-GP-UCB involve both HF and LF evaluations, while other methods only have HF evaluations. To ensure fair comparisons, among algorithms, the maximum number of the equivalent HF evaluations $\max_n E$ is limited to $(15d + 350)$ for all algorithms, where d is problem dimension. The number of equivalent HF evaluations is the sum of the HF evaluations and discounted LF evaluations. For example, if an algorithm has 100 LF evaluations and 10 HF evaluations, and since 10 LF evaluations equals 1 HF evaluation, then the total number of equivalent HF evaluations is $10 + 100/10 = 20$. Each of the algorithms is repeated for 30 independent trials and their average performances are compared.

For our proposed MRSO approach, the parameter setting for the DYCORS component is the same as the original DYCORS shown in Table 2. Here, we implement three different variants of the MRSO by using different optimizers (BFGS, GA, and multi-modal estimation of distribution algorithm (MEDA)) in the potential area detection procedure, named MRSO-BFGS, MRSO-GA, and MRSO-MEDA. The number of multi-trials on BFGS and GA is set to 100. For the BFGS, the algorithm will stop when the gradient norm is less than $1.0e - 05$. For GA, the population size is set to 100, the maximum number of iterations is set to 100, and the crossover and mutation

¹<https://github.com/dme65/pySOT>.

²<https://github.com/kirthevasank/mf-gp-ucb>.

Table 2 Parameter setting of compared algorithms

Algorithms	Parameters
SHPSO (Yu et al. 2018)	Swarm size= 50, constriction factor $\chi = 0.729$, $c_1 = c_2 = 2.05$, size of Latin hyper-cube sampling $n_s = 100$, size of data to train RBF surrogate $n_t = 100$
HSOSR (Dong et al. 2018)	Number of cheap points generated by LHS $n_c = 1000$, $n_{\text{rank1}} = 100$, $n_{\text{rank2}} = 50$, coefficient of distance $\Delta_1 = 1e-3$, $\Delta_2 = 1e-6$
S-JADE (Cai et al. 2019)	Swarm size= 30, initial value of $\mu_F = 0.5$ and $\mu_{CR} = 0.75$. $p = 0.05$ $r \in [0, 1.25]$, $L = 10$, type of RBF kernel is set to cubic.
DYCORS (Regis and Shoemaker 2013)	Size of initial experiment design $n_0 = 2(d+1)$, type of RBF kernel is set to cubic, Number of candidate points in each iteration $N_{\text{tri}} = 1000$, initial step size $\delta_{\text{init}} = 0.2$, maximum and minimum step sizes are $\delta_{\text{min}} = 0.005$ and $\delta_{\text{max}} = 0.2$, the tolerance for the number of consecutive failed and consecutive succeed iterations $\tau_f = \max(5, d)$ and $\tau_s = 3$. weight ω cycles between $\{0.1, 0.3, 0.5, 0.7, 0.9\}$
MGPMDE (Liu et al. 2016)	Threshold values of normalized population diversity $\delta_1 = 0.03$, $\delta_2 = 0.06$, number of groups in the pool $G = 5$, number of samples to represent each group $n_0 = 2$, size of seed population $\lambda_0 = 10$, population size of differential evolution $\lambda = 50$
MF-GP-UCB (Kandasamy et al. 2016)	Types of kernel used in GP model is set to squared exponential (SE), optimizer for selecting next evaluation point is set to dividing rectangles (DIRECT) (Jones et al. 1993), sizes of initial sampling points are set to $\lfloor \max_n E/10 \rfloor$ and $\lfloor \max_n E/100 \rfloor$ for LF and HF models, threshold value γ for switching fidelity is set to $(\lfloor 100 \cdot \text{range} Y \rfloor / 1.0e4)$ ($\text{range} Y$ is the gap between the maximum and minimum function values of the initial samples).

rate are set to 0.9 and $1/d$, respectively. For MEDA, the population size is set to 100, the maximum number of iterations is set to 200, and other parameter settings are taken from the original literature (Yang et al. 2016).

4.3 Impacts of different LF budget ratios on MRSO performance

One of the important issues in MFO is the allocation of the computational budgets between LF and HF runs. Too much search on LF model may cause an insufficient budget for further improving the solution on HF model. However,

too little search on LF model is also unwise because not enough of the cheaply obtained useful information from the LF model will be used. In this section, the impacts of different LF budget ratios (i.e., the percentage of the LF computational budget on the total computational budget) on MRSO performance are analyzed. We vary the LF budget ratio from 10 to 90% and run the three MRSO variants on the 10- and 30-dimensional test problems. Figure 4 depicts the trend of average algorithm performance over different LF budget ratio. The Y-axis values are the average final function values over 10 problems. Based on Fig. 4a, the MRSO algorithms tend to have better performance with

Fig. 4 Comparison on the average performance of MRSO variants with different LF budget ratio. **a** 10-dimensional problems. **b** 30 dimensional problems

Table 3 Mean and standard deviation of optimized HF function values found by each algorithm after 500 equivalent HF evaluations on 10-dimensional problems

Function	Metrics	HSOSR	SHPSO	S-JADE	MF-GP-UCB	MGPMDE	DYCORs	MRSO-GA	MRSO-BFGS	MRSO-MEDA
F ₁	Mean	6.51e+0	4.13e+0	4.39e+0	6.37e+0	1.56e+0	2.32e+0	1.83e+0	<i>1.29e+0</i>	1.43e+0
	Std	1.23e+0	1.30e+0	8.73e-1	1.23e+0	7.65e-1	1.09e+0	7.92e-1	<i>4.91e-1</i>	6.34e-1
F ₂	Mean	5.63e+0	3.57e+0	3.72e+0	6.02e+0	1.72e+0	1.87e+0	8.37e-1	9.71e-1	<i>6.17e-1</i>
	Std	1.25e+0	1.33e+0	8.64e-1	1.43e+0	1.33e+0	1.28e+0	7.82e-1	5.81e-1	<i>5.76e-1</i>
F ₃	Mean	6.51e+0	3.81e+0	4.39e+0	6.31e+0	1.92e+0	2.32e+0	1.42e+0	1.32e+0	<i>1.22e+0</i>
	Std	1.23e+0	1.44e+0	8.73e-1	1.42e+0	1.19e+0	1.09e+0	5.51e-1	<i>4.75e-1</i>	5.41e-1
F ₄	Mean	6.51e+0	3.97e+0	4.39e+0	6.93e+0	1.99e+0	2.32e+0	1.14e+0	<i>1.09e+0</i>	1.21e+0
	Std	1.23e+0	1.42e+0	8.73e-1	1.21e+0	7.77e-1	1.09e+0	<i>3.34e-1</i>	3.82e-1	4.01e-1
F ₅	Mean	6.51e+0	4.03e+0	4.59e+0	6.53e+0	1.47e+0	2.55e+0	1.29e+0	<i>9.64e-1</i>	1.24e+0
	Std	1.23e+0	1.22e+0	7.55e-1	1.25e+0	1.02e+0	1.21e+0	3.96e-1	<i>3.78e-1</i>	3.91e-1
F ₆	Mean	6.44e+0	3.99e+0	4.48e+0	6.50e+0	1.70e+0	2.58e+0	1.10e+0	<i>1.02e+0</i>	1.11e+0
	Std	1.39e+0	1.38e+0	9.92e-1	1.02e+0	1.06e+0	1.28e+0	4.91e-1	3.58e-1	<i>3.20e-1</i>
F ₇	Mean	6.51e+0	4.17e+0	4.59e+0	6.67e+0	1.80e+0	2.55e+0	9.40e-1	<i>8.76e-1</i>	8.86e-1
	Std	1.23e+0	1.60e+0	7.55e-1	1.18e+0	1.03e+0	1.21e+0	3.11e-1	<i>2.63e-1</i>	2.66e-1
F ₈	Mean	6.07e+0	4.39e+0	4.79e+0	6.41e+0	1.60e+0	2.56e+0	9.42e-1	<i>8.93e-1</i>	8.95e-1
	Std	1.02e+0	1.15e+0	9.23e-1	1.47e+0	9.08e-1	1.21e+0	<i>2.00e-1</i>	3.27e-1	3.49e-1
F ₉	Mean	6.06e+0	4.00e+0	4.33e+0	6.48e+0	2.01e+0	2.45e+0	1.30e+0	<i>1.19e+0</i>	1.35e+0
	Std	1.27e+0	1.21e+0	8.84e-1	1.09e+0	1.07e+0	1.29e+0	<i>4.08e-1</i>	5.06e-1	4.44e-1
F ₁₀	Mean	6.06e+0	4.01e+0	4.26e+0	5.70e+0	2.42e+0	2.41e+0	1.37e+0	1.37e+0	<i>1.29e+0</i>
	Std	1.27e+0	1.36e+0	9.23e-1	1.16e+0	1.17e+0	1.26e+0	<i>4.52e-1</i>	6.01e-1	4.97e-1

For each problem, the algorithms with the best mean and standard deviation values are in italics

Table 4 Mean and standard deviation of optimized HF function values found by each algorithm after 800 equivalent HF evaluations on 30-dimensional problems

Function	Metrics	HSOSR	SHPSO	S-JADE	MF-GP-UCB	MGPMDE	DYCORs	MRSO-GA	MRSO-BFGS	MRSO-MEDA
F ₁	Mean	2.85e+1	1.88e+1	2.38e+1	3.17e+1	9.97e+0	7.16e+0	6.45e+0	<i>5.84e+0</i>	6.62e+0
	Std	2.86e+0	2.76e+0	1.40e+0	1.50e+0	2.82e+0	2.33e+0	1.63e+0	<i>1.38e+0</i>	1.47e+0
F ₂	Mean	2.67e+1	1.67e+1	2.14e+1	2.90e+1	8.17e+0	4.50e+0	4.82e+0	<i>4.22e+0</i>	4.86e+0
	Std	1.83e+0	3.82e+0	2.02e+0	1.84e+0	3.20e+0	2.29e+0	1.66e+0	<i>1.30e+0</i>	1.75e+0
F ₃	Mean	2.85e+1	1.83e+1	2.38e+1	3.13e+1	9.81e+0	7.16e+0	6.43e+0	6.38e+0	<i>5.92e+0</i>
	Std	2.86e+0	3.06e+0	<i>1.40e+0</i>	1.70e+0	3.37e+0	2.33e+0	1.63e+0	1.58e+0	1.89e+0
F ₄	Mean	2.85e+1	1.87e+1	2.38e+1	3.13e+1	9.34e+0	7.16e+0	5.34e+0	5.15e+0	<i>5.09e+0</i>
	Std	2.86e+0	3.20e+0	1.40e+0	1.40e+0	2.69e+0	2.33e+0	1.39e+0	1.29e+0	1.43e+0
F ₅	Mean	2.85e+1	1.81e+1	2.32e+1	3.09e+1	8.48e+0	<i>6.86e+0</i>	7.37e+0	6.98e+0	7.17e+0
	Std	2.86e+0	2.97e+0	1.75e+0	1.70e+0	2.05e+0	2.10e+0	1.88e+0	2.18e+0	<i>1.39e+0</i>
F ₆	Mean	2.78e+1	1.75e+1	2.34e+1	3.15e+1	8.75e+0	6.82e+0	6.56e+0	<i>5.89e+0</i>	7.00e+0
	Std	3.08e+0	2.83e+0	1.62e+0	<i>1.59e+0</i>	2.10e+0	2.20e+0	1.93e+0	1.84e+0	1.72e+0
F ₇	Mean	2.85e+1	1.83e+1	2.32e+1	3.15e+1	9.11e+0	6.86e+0	<i>4.74e+0</i>	4.80e+0	4.92e+0
	Std	2.86e+0	2.81e+0	1.75e+0	1.61e+0	2.83e+0	2.10e+0	1.02e+0	<i>1.01e+0</i>	1.09e+0
F ₈	Mean	2.77e+1	1.81e+1	2.35e+1	3.21e+1	9.28e+0	7.03e+0	4.66e+0	4.64e+0	<i>4.42e+0</i>
	Std	2.34e+0	2.90e+0	1.49e+0	1.05e+0	2.56e+0	2.13e+0	9.97e-1	<i>7.02e-1</i>	8.59e-1
F ₉	Mean	2.83e+1	1.80e+1	2.32e+1	2.85e+1	8.81e+0	<i>6.48e+0</i>	6.58e+0	6.66e+0	6.70e+0
	Std	2.81e+0	3.13e+0	1.67e+0	2.81e+0	2.28e+0	1.99e+0	1.62e+0	1.75e+0	<i>1.39e+0</i>
F ₁₀	Mean	2.83e+1	1.84e+1	2.32e+1	3.15e+1	8.81e+0	<i>6.54e+0</i>	7.45e+0	7.46e+0	7.19e+0
	Std	2.81e+0	2.78e+0	2.56e+0	<i>9.54e-1</i>	2.61e+0	1.94e+0	1.41e+0	1.55e+0	1.76e+0

For each problem, the algorithms with the best mean and standard deviation values are in italics

mediate LF budget ratio for 10-dimensional problems. The trend becomes much apparent on 30-dimensional problems, and Fig. 4b shows that the performances are the best when LF budget ratio is 40%. Therefore, the LF budget ratio of 40% is used in the following comparison.

4.4 Comparison with other surrogate-assisted optimization methods

Tables 3 and 4 outline the statistical results achieved by all the compared methods including the mean values over

Fig. 5 Convergence curves of the alternative optimization methods on 10-dimensional benchmark functions (F_1 – F_6)

Fig. 6 Convergence curves of the alternative optimization methods on 10-dimensional benchmark functions (F_7 – F_{10})

30 trials and the standard deviations for 10- and 30-dimensional problems. It can be observed from Table 3 that both of the best mean and standard deviation are achieved by MRSO variants on the 10-dimensional problems. Among the three MRSO variants, the performance of MRSO-MEDA is slightly better than the other two MRSO variants. A similar situation can be observed from Table 4 where MRSO variants also outperform other methods on the 30-d problems except for F_2 . For F_2 , the original DYCORs has the best performance.

Figures 5, 6, 7, and 8 present the convergence profiles of all the compared methods on 10- and 30-dimensional problems. Recall that all the convergence curves of MRSO variants start from the 201th evaluation for 10-dimensional problems and 321th evaluation for 30-dimensional problems because the first 40% equivalent evaluations are spent on the LF models. The convergence curves show that all of the MRSO variants start from very low HF function values benefit from the sufficient exploration on the cheap LF models. For some problems, MRSO can achieve better solutions than other methods at the very beginning of the

HF run (i.e., F_7 and F_8 of the 10-dimensional problems, F_2 , F_3 , F_4 , F_7 , and F_8 of the 30-dimensional problems).

For the 10-dimensional problems, all of our three MRSO variants have superior convergence performance on 9 problems (F_2 to F_{10}). For F_1 , MRSO-BFGS and MRSO-MEDA have better convergence performance than other methods, and the performance of MRSO-GA is slightly worse than MGPMDE but better than the rest algorithms.

For the 30-dimensional problems, all of the three MRSO variants have superior convergence performance on 5 problems (F_1 , F_3 , F_4 , F_7 , F_8). For F_5 , F_6 , and F_{10} , the convergence performances of MRSO variants are worse than the original DYCORs. For F_2 and F_9 , the convergence performances of DYCORs and the MRSO are similar and both of them are better than the rest 5 methods.

4.5 Performance and data profiles

To have an overall performance evaluation on all the test problems for all compared algorithms, the performance and data profiles are used (Moré and Wild 2009; Regis 2014).

Fig. 7 Convergence curves of the alternative optimization methods on 30-dimensional benchmark functions. (F_1 – F_6)

The performance and data profiles provide an effective approach to analyze how well and how fast of a specific algorithm is at solving a set of problem instances. Let P be the set of problems, and p is one of the instances in P . Since there are 10 optimization problems on both 10- and 30-dimensional problems, and 30 independent

trials are repeated on each of the problems, there are $2 \times 10 \times 30 = 600$ problem instances to produce the performance and data profiles. In addition, let S be the set of compared algorithms. In this study, S contains 9 compared algorithms (HSOSR, SHPSO, S-JADE, BRF-DYCORS, MF-GP-UCB, MGPMDE, MRSO-BFGS, MRSO-GA, and

Fig. 8 Convergence curves of the alternative optimization methods on 30-dimensional benchmark functions (F_7 – F_{10})

MRSO-MEDA). For any pair (p, s) of a problem instance p and an algorithm s , the performance ratio is defined to be Moré and Wild (2009):

$$r_{p,s} = \frac{t_{p,s}}{\min(t_{p,s} : s \in S)} \quad (8)$$

where $t_{p,s}$ is the minimum number of function evaluations required by an algorithm to successfully pass the convergence test that is defined in Eq. 10. Clearly, $r_{p,s} \geq 1$ for any $p \in P$ and $s \in S$. Moreover, for problem p , the best algorithm s for this problem attains $r_{p,s} = 1$. Furthermore, when algorithm s fails to find a solution that satisfies the convergence test, then $r_{p,s} = \infty$.

For any algorithm $s \in S$ and $\alpha \geq 1$, the performance profile of algorithm s with respect to α shows the fraction over all problem instances where its performance ratio is no greater than α ,

$$\rho_s(\alpha) = \frac{1}{|P|} |p \in P : r_{p,s} \leq \alpha| \quad (9)$$

In this study, the convergence test proposed by More and Wild (2009) with a tolerance $\tau = 0.05$ is adopted. Let f_L be the minimum obtained function value among all the

algorithms on a particular problem within a given number of function evaluations, the convergence test for an algorithm s is passed if the following condition is satisfied:

$$f(\mathbf{x}_0) - f(\mathbf{x}) \geq (1 - \tau)(f(\mathbf{x}_0) - f_L) \quad (10)$$

where $\mathbf{x}_0$ is a starting search point on a particular problem instance. Here, the point with the worst fitness value among all the first evaluated points of all methods is selected as the starting search point.

Similarly, the data profile of an algorithm for the indicator of the optimization progress β (i.e., how much of the budget is consumed; β is in the range of [0.0%, 100.0%]), is defined by the following:

$$d_s(\beta) = \frac{1}{|P|} |p \in P : t_{p,s} \leq \beta| \quad (11)$$

where $t_{p,s}$ is the minimum number of objective function evaluations required by an algorithm to successfully pass the convergence test on problem p . For any algorithm $s \in S$, the data profile curve of s shows its growth of successful rate over all problem instances with the increasing of β .

Figure 9 shows the performance and data profiles. Note that for both performance profiles and data profiles the

Fig. 9 **a** Performance and **b** data profiles on all the problems (LF budget ratio = 40%, and highest curves are the best)

best results are the highest curves. The data profiles are especially important for expensive optimization problems because the horizontal axis indicate the optimization progress (it is positively correlated to the consumed number of equivalent HF function evaluations). Figure 9 a shows that the MRSO variants, followed by the original DYCORDS and MGPMDE, have dramatically better performance profiles than other methods. Similarly, as shown by Fig. 9b, MRSO variants have better data profile values than other methods. In particular, the performance profile curves indicate that MRSO variants obtain the best solutions for about 72% to 80% out of 600 problem instances. The corresponding percentage of the original DYCORDS is about 46%, and the corresponding percentage for MGPMDE is below 40%. For the remaining 4 methods (HSOSR, SHPSO, S-JADE, and MF-GP-UCB), the corresponding percentage is close to zero, which means they failed to pass the convergence tests. The data profile curves indicate that the MRSO variants obtain better performance than other methods since 50% of the whole optimization process. Notice that MGPMDE has better performance at the earlier stage of the optimization process but drops behind DYCORDS and MRSO variants at the later stage. It is probably because that MGPMDE is trapped in local optimums and cannot further improve the solutions while

DYCORDS and MRSO variants have the restart strategy so that they are more likely to jump out the local optima.

4.6 Discussion

According to the impact analysis of LF budget ratio to MRSO's performance in Fig. 4, it is sensible to use about 40% of the computational budget for the search on LF model. The LF budget ratio of 40% is also used in the comparison to other methods. It should be recalled that the results in Fig. 9 for all methods assume that the maximum computational budget and the percentage of budget to be used for LF evaluations is fixed in advance of any optimization. Hence, one should not interpret the results in Fig. 9b to indicate that other methods outperform MRSO options when the evaluation budget is less. What Fig. 9b indicates is that MRSO is using (the less accurate) LF evaluation in those early iterations in contrast to DYCORDS (that uses no LF computation) so it is not surprising the those objective values are lower in early computation. Also, MRSO methods would change their search method (which is a function of the computational budget allocated, see (4)) if the maximum evaluation effort were reduced. Hence, the fact that curves in Fig. 9b are lower than the MRSO for some other algorithms when fewer than 50%

Fig. 10 **a** Performance and **b** data profiles on all the problems. (LF budget ratio = 10%, and highest curves are the best)

Fig. 11 A two-station tandem queue

of the evaluation budget is expended just indicates MRSO is searching differently in early evaluations than the other methods. The valid number to look at is the performance at 100% of budget, where the MRSO algorithms have excellent performance in comparison to others.

It is also interesting to know if MRSO variants still outperform other methods when using other settings on the LF budget ratio. Hence, we use the results of MRSO variants with LF budget ratio of 10% to plot the performance and data profiles again. Interestingly, it can be observed from Fig. 10a, although there is a drop in the performance of MRSO variants, they still outperform other methods. In addition, it can be seen from the data profiles in Fig. 10b that MRSO variants keep ahead from the very beginning of the optimization process. Therefore, it demonstrates the high robustness of MRSO's performance.

5 Applications on capacity planning for manufacturing systems

To further demonstrate the effectiveness of the proposed MRSO, in this section, we apply MRSO to the capacity planning problem in manufacturing systems. Capacity planning is of critical importance because investment in facility is very high in manufacturing systems. For example, the cost of new wafer fabs in a semiconductor factory can be more than US\$1.0 billion (Hopp et al. 2002). A manufacturing system usually consists of different types of machines, and the managers have to determine the optimal capacity of each type of machine to minimize the system cost. This problem is challenging because the system performance is analytically intractable and accurate performance analysis of the system relies on running expensive simulation models. In other words, the problem can be recognized as a computationally expensive black-box optimization problem. In the following, we will introduce the detailed problem settings and compare the performance of the proposed MRSO with other algorithms in the literature.

5.1 The capacity planning problem

For simplicity, we model the manufacturing system with a queuing network and follow the outline given in Hopp et al. (2002). Jobs (or products) arrive to the system with the rate λ and need to be processed by the machines based

on a deterministic sequence of procedures. Note that a product can visit one station multiple times (i.e., re-entry allowed) (Kumar 1993). There are N types of machines in the system and capacity planning involves determining the optimal capacity for these machines. We denote the decision variables, i.e., the capacity of each type of machine, by $\mu = (\mu_1, \mu_2, \dots, \mu_N)$. The objective function (i.e., system cost) is comprised of two parts: the capacity cost and waiting cost. The capacity cost (CC) includes the machine purchasing operations cost and is an increasing function with respect to each μ_i , $i = 1, 2, \dots, N$. The waiting cost (WC) is the function of jobs' average cycle times (CT) in the manufacturing system. The reason for introducing the WT is that the CT can represent the service level and large cycle time generally means large volume of work in process (Little's Law, Little and Graves (2008)), which will induce inventory cost. Depending on the practical situations, the relationship between the WT and CT can have different forms. CT depends on $(\mu_1, \mu_2, \dots, \mu_N)$ and its accurate evaluation relies on the simulation model for the manufacturing systems.

Hence, the capacity planning problem can be formulated as follows:

$$\min_{\mu \in \mathcal{D}} f(\mu) = \text{WC}(\text{CT}(\mu)) + \text{CC}(\mu). \quad (12)$$

Here, $\mathcal{D}$ is the feasible solution space. For stability, the capacity of each type of station should be greater than the effective arrival rate at the station (Shen and Wu 2018; Hopp et al. 2002). Note that $f(\mu)$ is a black-box function since the evaluation for $\text{CT}(\mu)$ relies on expensive simulation models.

In this context, an HF model means we run the simulation model for a sufficiently long period to get an accurate estimate for the average cycle and then the objective function can be calculated accordingly. The LF model may be created in different ways. For example, one can employ the approximation models in queuing theory to estimate $\text{CT}(\mu)$ (Whitt 1983; Wu and McGinnis 2013; Wu et al. 2018). These approximations are inaccurate but their computation is very inexpensive. Alternatively, an LF model can also be created by running the simulation model with less computing effort, which will provide an inaccurate estimate for $\text{CT}(\mu)$. To be consistent with the numerical studies in previous sections, we use the latter method as the LF model.

5.2 An illustration example of two stations

To better visualize the HF and LF models in a manufacturing system, we first present an example consisting of only two types of stations (i.e., a 2-d problem). As shown by Fig. 11, jobs arrive from outside with rate $\lambda = 1$ and need to be processed by station 1 and station 2 sequentially. The CC

Fig. 12 Comparison between the LF and HF models on the two-station tandem queue problem

is assumed be of the form $CC(\mu) = c_1(\mu_1 - 1.2)^3 + c_1(\mu_1 - 1.2)^3$. The waiting cost is assumed to be proportional to the average cycle time $WC = hCT(\mu_1, \mu_2)$.

Under the above setting, it follows that

$$f(\mu_1, \mu_2) = c_1(\mu_1 - 1.2)^3 + c_2(\mu_2 - 1.2)^3 + hCT(\mu_1, \mu_2), \quad (13)$$

where $c_1 = 3500$, $c_2 = 4000$, and $h = 0.2$ are the cost parameters. For stability, the domain of (μ_1, μ_2) is $(1, 1.4] \times (1, 1.4]$. We assume Gamma distributions for the

inter-arrival time of jobs and service times at each stations. The squared coefficient of variation (SCVs) of the arrival process, service at station 1, and service at station 2 are 10, 0.1, 1.5 respectively.

To visualize the similarities and differences between the LF model and HF model, we produce the surface plot and contour plot of $f(\mu_1, \mu_2)$ for the two models respectively. As shown in Fig. 12, although there is a performance gap between the LF model and HF model, they share a similar structure. Hopefully, we can exploit this property to improve the efficiency of determining the optimal capacity.

Table 5 Processing procedures of jobs

Step	1	2	3	4	5	6	7	8	9	10
Station	Preclean	Laser	Alignment	Clean	Photo	Etch	Strip	Oxide	Mask	Photo
Step	11	12	13	14	15	16	17	18	19	20
Station	Etch	Strip	Clean	Nitride	Mask	Photo	Etch	Strip	Poly	Probe

Table 6 Capacity cost information at each station and feasible space for each decision variable

Index	Station	Variable	Unit cost (\$, 000)	Domain
1	Preclean	μ_1	65/3	(5.05, 15.15)
2	Laser	μ_2	220/3	(5.05, 15.15)
3	Alignment	μ_3	175/6	(5.05, 15.15)
4	Clean	μ_4	110/3	(10.10, 30.30)
5	Photo	μ_5	350/3	(15.15, 45.45)
6	Etch	μ_6	15/2	(15.15, 45.45)
7	Strip	μ_7	28/3	(15.15, 45.45)
8	Oxide	μ_8	175/6	(5.05, 15.15)
9	Mask	μ_9	65/4	(10.10, 30.30)
10	Nitride	μ_{10}	55/3	(5.05, 15.15)
11	Poly	μ_{11}	15	(5.05, 15.15)
12	Probe	μ_{12}	15	(5.05, 15.15)

5.3 Experimental results on a 12-*d* capacity planning problem

In this section, the proposed MRSO is applied to a 12-*d* capacity planning problem, and we compare its performance with other methods (the same methods in Section 4). We use a simplified version of the example given by Hopp et al. (2002). We consider a semiconductor line with 12 types of machines and 1 type of job. Jobs arrive to the system with the rate $\lambda = 1$, and need to be processed based a deterministic sequence of procedures given in Table 5. In this case, the system cost is assumed to take the following form:

$$f(\mu) = \sum_{i=1}^{i=12} c_i \mu_i + e^{hCT(\mu)}, \quad (14)$$

where c_i is the unit cost for each type of machine and h is a constant. In other words, the CC is proportional to μ and the waiting cost increases exponentially with respect to the average cycle time $CT(\mu)$. The CC information and feasible space of each decision variable is given in Table 6 and the constant h is set as 2.

To evaluate the average cycle time and then calculate the objective function, the HF model simulates the manufacturing system for 2 million hours and the LF model simulates the system for 0.2 million hours. A single run of

Fig. 13 Progress plots of the compared methods on the capacity planning problem

the HF model takes about 30 min and a single run of the LF model takes about 3 min. Hence, the computational cost ratio between HF and LF model is also 10.0. Since the HF evaluation is expensive, thus we set the maximum number of equivalent HF evaluation as 150 and each of the compared methods is repeated for 10 trials. Each run of the whole optimization process (with 150 equivalent HF evaluations) takes more than 3 days of computing time.

Table 7 presents the statistic results from 10 trials of all the compared methods on this problem. The results show that MRSO-BFGS obtains the best results. Figure 13 presents the progress plots of all the methods on the capacity planning problem. It can be found that the MRSO variants find excellent solutions within a small number of evaluations. In addition, compared with the original DYCORDS, the MRSO variants, especially MRSO-GA and MRSO-BFGS, converge to the optimal solution much faster. This validates that the proposed MFO framework can further speed-up the optimization process compared with the original DYCORDS. In summary, the superior performance of MRSO variants in the capacity planning problem has demonstrated its potential to efficiently solve practical expensive optimization problems.

Table 7 Statistics of the results obtained by different methods on the 12-*d* capacity planning problem. The best value on each metric is in italic

Metrics	HSOSR	SHPSO	S-JADE	MF-GP-UCB	MGPMDE	DYCORS	MRSO-GA	MRSO-BFGS	MRSO-MEDA
Best	4.14e+3	4.32e+3	4.31e+3	4.01e+3	4.15e+3	4.02e+3	4.03e+3	<i>3.98e+3</i>	4.04e+3
Worst	4.19e+3	4.56e+3	4.93e+3	4.16e+3	4.17e+3	4.19e+3	4.16e+3	<i>4.15e+3</i>	4.20e+3
Median	4.18e+3	4.38e+3	4.65e+3	4.12e+3	4.16e+3	4.08e+3	4.07e+3	<i>4.03e+3</i>	4.10e+3
Mean	4.18e+3	4.39e+3	4.64e+3	4.11e+3	4.16e+3	4.10e+3	4.08e+3	<i>4.05e+3</i>	4.10e+3
Std	1.59e+1	7.27e+1	1.97e+2	4.76e+1	<i>4.51e+0</i>	5.47e+1	3.81e+1	6.42e+1	4.31e+1

6 Conclusions

In this paper, a multi-fidelity RBF surrogate-based optimization framework called MRSO is developed for computationally expensive black-box optimization problems. The proposed MRSO approach has three main steps: (1) search on the LF model, (2) potential area detection procedure, and (3) the search on the HF model. The proposed MRSO approach is tested and compared with other 6 recent surrogate-based optimization methods on 10 test problems from the MFO test suite with 10 and 30 dimensions. Experimental results show that MRSO can speed-up the optimization process and further improve the performance of the original DYCORDS algorithm. Furthermore, it outperforms other compared single-fidelity and multi-fidelity surrogate-based optimization methods. Besides the results on test functions, MRSO is successfully applied to a complex, computationally expensive real-world capacity planning problem in manufacturing systems involving discrete event simulations. Therefore, the proposed MRSO can be considered as a promising approach for solving real-life computationally expensive optimization problems.

In the future, we will apply the proposed MRSO to other complex, computationally expensive optimization problems in real life.

Acknowledgments This work has been funded by Phase 2 of Energy and Environmental Sustainability Systems (E2S2) CREATE project and the Institute for Operations Research Analytics, both at National University of Singapore.

The computational work for this article was partially performed at the National Supercomputing Centre, Singapore.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results The experimental results from this work can be reproduced by using the pySOT package in Github (Eriksson et al. 2019). The HF and LF model of the 12-*d* capacity planning problem can be found in the Supplementary Material.

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/>.

Appendix

Table 8 List of notation used in the proposed MRSO

Name	Description
$\mathcal{D}_L$	Sampling set of the Latin hyper-cube design for LF model
n_0	Number of sampling points in the Latin hyper-cube design for LF model
$f_L(\mathbf{x})$	Objective function value on LF model at $\mathbf{x}$
n_L	Number of evaluations on LF model
n_H	Number of evaluations on HF model
$\max n_L$	Maximum allowed number of LF evaluations in LF run
$\max n_E$	Maximum allowed number of equivalent HF evaluations
N_{can}	Number of candidate points generated in each iteration in LF and HF run
C_L	Set of candidate points generated in each iteration in LF run
C_H	Set of candidate points generated in each iteration in HF run
$\mathbf{x}_{\text{nl}}$	The selected next evaluation point in LF run
$\mathbf{x}_{\text{nh}}$	The selected next evaluation point in HF run
R	Computational cost ratio between HF and LF models
Θ_L	Database for storing the evaluated x-y pairs on LF model
Θ_H	Database for storing the evaluated x-y pairs on HF model
$\mathbf{x}_L^*, f_{\text{best}}^L$	Best solution found so far on LF model and its function value
$\mathbf{x}_H^*, f_{\text{best}}^H$	Best solution found so far on HF model and its function value
$S_L(\mathbf{x})$	RBF surrogate on LF model
$S_H(\mathbf{x})$	RBF surrogate on HF model
$\mathcal{A}$	Set for storing the final solutions after the search on LF surrogate
N_g	Number of cluster groups
$Q\%$	The percentage of top points from the obtained points in step 4.2
N_{tri}	Number of candidate points generated in each iteration
γ	Cycling pattern for selecting next evaluation point

References

- Ali W, Khan MS, Qyyum MA, Lee M (2018) Surrogate-assisted modeling and optimization of a natural-gas liquefaction plant. *Comput Chem Eng* 118:132–142
- Battiti R, Masulli F (1990) BFGS optimization for faster and automated supervised learning. In: *International neural network conference*, Springer, pp 757–760
- Bonfiglio L, Perdikaris P, Brizzolara S, Karniadakis G (2018) Multi-fidelity optimization of super-cavitating hydrofoils. *Comput Methods Appl Mech Eng* 332:63–85
- Cai X, Gao L, Li X, Qiu H (2019) Surrogate-guided differential evolution algorithm for high dimensional expensive problems. *Swarm Evol Comput* 48:288–311
- Ding F, Kareem A (2018) A multi-fidelity shape optimization via surrogate modeling for civil structures. *J Wind Eng Ind Aerodyn* 178:49–56
- Dong H, Song B, Wang P, Dong Z (2018) Hybrid surrogate-based optimization using space reduction (HSOSR) for expensive black-box functions. *Appl Soft Comput* 64:641–655
- Durantin C, Rouxel J, Désidéri JA, Glière A (2017) Multifidelity surrogate modeling based on radial basis functions. *Struct Multidiscip Optim* 56(5):1061–1075
- Eriksson D, Bindel D, Shoemaker CA (2019) pySOT and POAP: an event-driven asynchronous framework for surrogate optimization. [arXiv:190800420](https://arxiv.org/abs/190800420)
- Friedman JH et al (1991) Multivariate adaptive regression splines. *Ann Stat* 19(1):1–67
- Habib A, Singh HK, Ray T (2019) A multiple surrogate assisted multi/many-objective multi-fidelity evolutionary algorithm. *Inform Sci* 502:537–557
- Haftka RT, Villanueva D, Chaudhuri A (2016) Parallel surrogate-assisted global optimization with expensive functions—a survey. *Struct Multidiscip Optim* 54(1):3–13
- Han ZH, Chen J, Zhang KS, Xu ZM, Zhu Z, Song WP (2018) Aerodynamic shape optimization of natural-laminar-flow wing using surrogate-based approach. *AIAA J* 56(7):2579–2593
- Hennig P, Schuler CJ (2012) Entropy search for information-efficient global optimization. *J Mach Learn Res* 13:1809–1837
- Hopp WJ, Spearman ML, Chayet S, Donohue KL, Gel ES (2002) Using an optimized queueing network model to support wafer fab design. *Iie Trans* 34(2):119–130
- Hutter F, Hoos HH, Leyton-Brown K, Murphy KP (2009) An experimental investigation of model-based parameter optimisation: SPO and beyond. In: *Proceedings of the 11th annual conference on Genetic and evolutionary computation*, ACM, pp 271–278
- Ilievski I, Akhtar T, Feng J, Shoemaker CA (2017) Efficient hyperparameter optimization for deep learning algorithms using deterministic rbf surrogates. In: *31st AAAI conference on artificial intelligence*, pp 1–8
- Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm Evol Comput* 1(2):61–70
- Jones DR, Pertunen CD, Stuckman BE (1993) Lipschitzian optimization without the Lipschitz constant. *J Optim Theory Appl* 79(1):157–181
- Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Global Optim* 13(4):455–492
- Kandasamy K, Dasarathy G, Oliva JB, Schneider J, Póczos B (2016) Gaussian process bandit optimisation with multi-fidelity evaluations. In: *Advances in neural information processing systems*, pp 992–1000
- Kennedy J (2010) Particle swarm optimization. *Encyclopedia of machine learning* pp 760–766
- Kenny QY, Li W, Sudjianto A (2000) Algorithmic construction of optimal symmetric Latin hypercube designs. *J Stat Plan Inference* 90(1):145–159
- Koziel S, Leifsson L, Yang XS (2014) *Solving computationally expensive engineering problems: methods and applications*, vol 97. Springer, Berlin
- Kumar P (1993) Re-entrant lines. *Queueing Syst* 13(1-3):87–110
- Li C, de Celis Leal DR, Rana S, Gupta S, Sutti A, Greenhill S, Slezak T, Height M, Venkatesh S (2017) Rapid Bayesian optimisation for synthesis of short polymer fiber materials. *Sci Report* 7(1):5683
- Little JD, Graves SC (2008) Little's law. In: *Building intuition*, Springer, pp 81–100
- Liu B, Koziel S, Zhang Q (2016) A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems. *J Comput Sci* 12:28–37
- Liu B, Yang H, Lancaster MJ (2017) Global optimization of microwave filters based on a surrogate model-assisted evolutionary algorithm. *IEEE Transactions on Microwave Theory and Techniques* 65(6):1976–1985
- Liu H, Ong YS, Cai J, Wang Y (2018) Cope with diverse data structures in multi-fidelity modeling: a Gaussian process method. *Eng Appl Artif Intel* 67:211–225
- Liu Z, Bhattacharjee KS, Tian FB, Young J, Ray T, Lai JC (2019) Kinematic optimization of a flapping foil power generator using a multi-fidelity evolutionary algorithm. *Renew Energy* 132:543–557
- Lu D, Ricciuto D, Stoyanov M, Gu L (2018) Calibration of the E3SM land model using surrogate-based global optimization. *J Adv Model Earth Syst* 10(6):1337–1356
- Moré JJ, Wild SM (2009) Benchmarking derivative-free optimization algorithms. *SIAM J Optim* 20(1):172–191
- Müller J, Shoemaker CA (2014) Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *J Glob Optim* 60(2):123–144
- Powell M (1992) *The theory of radial basis function approximation in 1990*, *Advances in Numerical Analysis II: Wavelets, Subdivision, and Radial Functions* (WA Light, ed.) Oxford University Press, Oxford, p 105:210
- Redouane K, Zeraibi N, Amar MN (2019) Adaptive surrogate modeling with evolutionary algorithm for well placement optimization in fractured reservoirs. *Appl Soft Comput* 80:177–191
- Regis RG (2014) Particle swarm with radial basis function surrogates for expensive black-box optimization. *J Comput Sc* 5(1):12–23
- Regis RG, Shoemaker CA (2007) A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J Comput* 19(4):497–509
- Regis RG, Shoemaker CA (2013) Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Eng Optim* 45(5):529–555
- Schubert E, Sander J, Ester M, Kriegel HP, Xu X (2017) DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Trans Database Syst (TODS)* 42(3):19
- Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N (2015) Taking the human out of the loop: a review of Bayesian optimization. *Proc IEEE* 104(1):148–175
- Shen Y, Wu K (2018) Stability of a $GI/G/1$ queue: a survey. *Asia-Pac J Oper Res* 35(03):page 1850015
- Simpson TW, Booker AJ, Ghosh D, Giunta AA, Koch PN, Yang RJ (2004) Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Struct Multidiscip Optim* 27(5):302–313
- Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. *Stat Comput* 14(3):199–222

- Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. In: *Advances in neural information processing systems*, pp 2951–2959
- Song X, Lv L, Sun W, Zhang J (2019) A radial basis function-based multi-fidelity surrogate model: exploring correlation between high-fidelity and low-fidelity models. *Struct Multidiscip Optim* 60(3):965–981
- Tao J, Sun G (2019) Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization. *Aerosp Sci Technol* 92:722–737
- Villarrubia G, De Paz JF, Chamoso P, De la Prieta F (2018) Artificial neural networks used in optimization problems. *Neurocomputing* 272:10–16
- Wang H, Jin Y (2018) A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems. *IEEE Trans Cybern* 50(2):536–549
- Wang H, Jin Y, Doherty J (2017) A generic test suite for evolutionary multifidelity optimization. *IEEE Trans Evol Comput* 22(6):836–850
- Whitley D (1994) A genetic algorithm tutorial. *Stat Comput* 4(2):65–85
- Whitt W (1983) Performance of the queueing network analyzer. *Bell Syst Tech J* 62(9):2817–2843
- Wild SM, Shoemaker C (2013) Global convergence of radial basis function trust-region algorithms for derivative-free optimization. *SIAM Rev* 55(2):349–371
- Wu K, McGinnis L (2013) Interpolation approximations for queues in series. *IIE Trans* 45(3):273–290
- Wu K, Srivathsan S, Shen Y (2018) Three-moment approximation for the mean queue time of a GI/G/1 queue. *IIE Trans* 50(2):63–73
- Yang Q, Chen WN, Li Y, Chen CP, Xu XM, Zhang J (2016) Multimodal estimation of distribution algorithms. *IEEE Trans Cybern* 47(3):636–650
- Yi J, Gao L, Li X, Shoemaker CA, Lu C (2019) An on-line variable-fidelity surrogate-assisted harmony search algorithm with multi-level screening strategy for expensive engineering design optimization. *Knowl-Based Syst* 170:1–19
- Yong HK, Wang L, Toal DJ, Keane AJ, Stanley F (2019) Multi-fidelity Kriging-assisted structural optimization of whole engine models employing medial meshes. *Struct Multidiscip Optim* 60(3):1209–1226
- Yu H, Tan Y, Zeng J, Sun C, Jin Y (2018) Surrogate-assisted hierarchical particle swarm optimization. *Inform Sci* 454:59–72
- Zhang H, Li Y, Chen Z, Su X, Yuan X (2019a) Multi-fidelity model based optimization of shaped film cooling hole and experimental validation. *Int J Heat Mass Transf* 132:118–129
- Zhang J, Xiao M, Gao L, Chu S (2019b) A combined projection-outline-based active learning Kriging and adaptive importance sampling method for hybrid reliability analysis with small failure probabilities. *Comput Methods App Mech Eng* 344:13–33
- Zhang Y, Han ZH, Zhang KS (2018) Variable-fidelity expected improvement method for efficient global optimization of expensive functions. *Struct Multidiscip Optim* 58(4):1431–1451
- Zheng J (2018) An output mapping variable fidelity metamodeling approach based on nested latin hypercube design for complex engineering design optimization. *Appl Intell* 48(10):3591–3611
- Zheng J, Shao X, Gao L, Jiang P, Qiu H (2015) Difference mapping method using least square support vector regression for variable-fidelity metamodelling. *Eng Optim* 47(6):719–736
- Zhou Q, Wang Y, Choi SK, Jiang P, Shao X, Hu J (2017) A sequential multi-fidelity metamodeling approach for data regression. *Knowl-Based Syst* 134:199–212
- Zhu C, Byrd RH, Lu P, Nocedal J (1997) Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans Math Softw (TOMS)* 23(4):550–560

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