

# Adaptive efficient global optimization of systems with independent components

Samee Ur Rehman<sup>1</sup> · Matthijs Langelaar<sup>1</sup>

Received: 9 November 2015 / Revised: 1 January 2017 / Accepted: 26 January 2017 / Published online: 18 February 2017  
© The Author(s) 2017. This article is published with open access at Springerlink.com

**Abstract** We present a novel approach for efficient optimization of systems consisting of expensive to simulate components and relatively inexpensive system-level simulations. We consider the types of problem in which the components of the system problem are independent in the sense that they do not exchange coupling variables, however, design variables can be shared across components. Component metamodels are constructed using Kriging. The metamodels are adaptively sampled based on a system level infill sampling criterion using Efficient Global Optimization. The effectiveness of the technique is demonstrated by applying it on numerical examples and an engineering case study. Results show steady and fast converge to the global deterministic optimum of the problems.

**Keywords** System optimization · Kriging · Expected improvement · Efficient global optimization · Infill sampling · Gaussian processes

## 1 Introduction

Optimization of systems has been dealt with in different ways depending on the problem type. The field of multidisciplinary design optimization has evolved at a fast rate in the past decades with the development of various methods and

techniques to address optimization of systems with many components or disciplines. In multidisciplinary design optimization the primary focus has been on optimization of systems involving multiple interactions between disciplines which are coupled with one another via coupling variables. Various techniques have been developed that aid in maintaining consistency between these disciplines during the optimization process (Martins and Lambe 2013).

A majority of systems falls under the category of a problem where multiple components have complex interactions with one another. However, a subset of engineering and non-engineering systems consists of components that are not interdependent. Such systems may contain several components but the response of these components can be evaluated independently of one another. In other words, the bi-level problem has a hierarchical structure where, at the lower level, each component in the system is only a function of design variables and there is no exchange of coupling variables between components. In other words, the input-output relation of a component is not influenced by the response of any other components. For this type of problem there is no requirement for maintaining consistency between the different disciplines or components and the system optimization process is therefore much simpler.

In addition, components could be expensive to evaluate but the system transformation may be cheap. Systems of this type are common in the field of integrated optics (Lifante 2003). The response of integrated optical components such as power splitters, directional couplers and phase shifters can be obtained independently of other components. Finding the component response usually requires computationally expensive electromagnetic simulations. Multiple such components can be used to develop complex integrated optical systems. An example of such a system is an integrated optical serial ring resonator (Ahmed et al. 2011).

---

✉ Matthijs Langelaar  
M.Langelaar@tudelft.nl

<sup>1</sup> Precision and Microsystems Engineering, Delft University of Technology, Delft, The Netherlands

This work addresses optimization of deterministic hierarchical systems based on independent components that are expensive to evaluate. It is assumed that, once the component responses are available, the system response is cheap to compute. In order to optimize the system we construct response surfaces of the underlying components. The accuracy of the response surfaces depends on the number of samples and their locations. The challenge is to perform the optimization with a desired accuracy at very low computational cost. The response surfaces are built using Kriging (Sacks et al. 1989). Kriging provides an estimator of the Mean Squared Error (MSE) in its interpolation. This MSE was used by Jones et al. (1998) to develop the Expected Improvement (EI) criterion in Efficient Global Optimization (EGO). The EI criterion enables adaptive sampling of unconstrained problems in order to estimate the optimum using relatively few expensive simulations.

We develop a system level EI criterion that is derived from a *system level* MSE. The system MSE is found by performing a linear transformation of the component level MSE generated by each component metamodel. The system level EI suggests a potential location to be adaptively sampled at each iteration of the algorithm and the component metamodelling are updated at the corresponding location. The iterative update of the component metamodelling results in a higher fidelity system response with each passing iteration. The process enables the optimum of the system to be estimated using relatively few component simulation calls.

Metamodels have often been used in multidisciplinary optimization (Viana et al. 2014). Improving systems by adaptively sampling component metamodels is also a fairly well known concept (Li et al. 2006; Hu et al. 2011). Optimization of multilevel decomposed systems that have a hierarchical structure has also received attention (Kokkolaras et al. 2004). Similarly, (Barton 1997) employed different strategies for component metamodel construction and update for system level optimization. Research has also been previously performed on partitioning expensive problems with a large number of variables so that they can be treated using component metamodels in a multilevel system framework (Koch et al. 2000), Haftka2005. On the other hand, (Xiu et al. 2013) specifically studied the quantification of error at system level due to component level metamodels in a hierarchical system.

This work also aims to optimize a multi-level problem with a hierarchical structure. The bi-level problem consists of a cheap system transformation at the upper level and expensive component models at the lower level. To our knowledge, there has been no previous work on a system level *expected improvement* based sampling strategy for optimization of systems involving expensive to simulate components that are independent of each other. The

proposed approach can efficiently optimize such systems, provided the number of component variables is so small that high fidelity component metamodels can be built using a reasonable number of simulations.

The method is especially relevant for systems in which the component behavior is easier to approximate than system behavior. Also, it may be advantageous to employ the technique in situations where many similar components are present in the system since a single metamodel can then often replace multiple components. Furthermore, the approach is attractive for fields in which the systems are typically composed of a selection of a small set of components.

The paper is organized as follows. The system level optimization problem of systems with independent components is described in Section 2. In Section 3, we introduce Kriging and EGO and then present the system level EI approach for adaptive sampling of the system. The algorithm is tested on several well known numerical test problems and the result is analyzed in Section 4. The application of the algorithm on a practical engineering problem is shown in Section 5. Finally, Section 6 contains the conclusions and suggestions for future work.

## 2 Optimization of systems with independent components

We introduce unconstrained optimization of the system in this section, but the method also applies to constrained problems. The constrained problem will be discussed in Section 3.1.3.

Let  $S$  represent the response of a system of a set of  $\mathcal{N}$  uncoupled components  $\mathbf{c}$ . Each component  $c_i$ , where  $\{i | i \in \mathbb{N}, i \leq \mathcal{N}\}$ , is a function of design variables  $\mathbf{x}_i \in \mathbb{X}_i \subseteq \mathbb{X}$ . Since design variables could potentially be shared across components, the sets  $\mathbb{X}_1$  to  $\mathbb{X}_{\mathcal{N}}$  may or may not be disjoint. In addition, some design variables  $\mathbf{x}_s \in \mathbb{X}_s$  may only be present at system level, or may be present at both component and system level. The union of sets  $\mathbb{X}_i \forall i \in \mathcal{N}$  and  $\mathbb{X}_s$  is the set  $\mathbb{X}$ . We define design variables that are only present in a single component as local variables. Design variables that are present in multiple component or at both component and system level are referred to as global variables. The optimization of the system response can be written as,

$$\min_{\mathbf{x} \in \mathbb{X}} S(c_1(\mathbf{x}_1), c_2(\mathbf{x}_2), c_3(\mathbf{x}_3), \dots, c_{\mathcal{N}}(\mathbf{x}_{\mathcal{N}}), \mathbf{x}_s). \quad (1)$$

If the system response is based on expensive to simulate components, then applying optimization directly on the system response is prohibitively costly. A more viable option instead is to construct lower dimensional (i.e. cheaper)

metamodels  $\mathcal{K}_{c_i}$  of the components and to apply optimization on the resulting system,

$$\min_{\mathbf{x} \in \mathbb{X}} S_{\mathcal{K}}(\mathcal{K}_{c1}(\mathbf{x}_1), \mathcal{K}_{c2}(\mathbf{x}_2), \dots, \mathcal{K}_{cN}(\mathbf{x}_N), \mathbf{x}_s). \tag{2}$$

The optimization problem in (2) is visualized in Fig. 1a. In this work, we construct the metamodels using Kriging which provides an estimator of the error in its interpolation. We then adaptively sample the component metamodels in such a way that the system optimum is found using only a limited number of expensive simulations of the components.

The fidelity of the system response that is based on component metamodels will heavily depend on the error of the underlying component response surfaces. Furthermore, the relative amount of system error is also governed by the operation that is performed on the component responses in order to arrive at the system response. This operation could turn a small component error into a large error contribution on the system level and vice versa. Therefore, it is important that any adaptive sampling of the components takes into account the error generated at system level. In the following section we show how such an adaptive sampling scheme can be developed.

### 3 Infill sampling criterion for system level optimization

#### 3.1 Kriging and efficient global optimization

##### 3.1.1 Kriging

Kriging is a metamodeling method that constructs the best linear unbiased predictor through a set of responses based

on certain statistical assumptions. The most important of these assumptions is that the function responses (outputs) at two input locations (points) are more correlated, the closer these points are. Kriging is also a popular method for constructing response surfaces of fully deterministic simulation results (Forrester et al. 2008). Before introducing our new method in Section 3.2, we provide a condensed explanation of the main steps in Kriging metamodel construction. Some equations are included in the Appendix, for reference. Sacks et al. (1989) provide a full description of Kriging.

Kriging uses a tunable basis function which usually takes the form of a parameterized Gaussian correlation function, (3), to measure the correlation between the outputs at sample points. This correlation between these outputs is given by the following expression

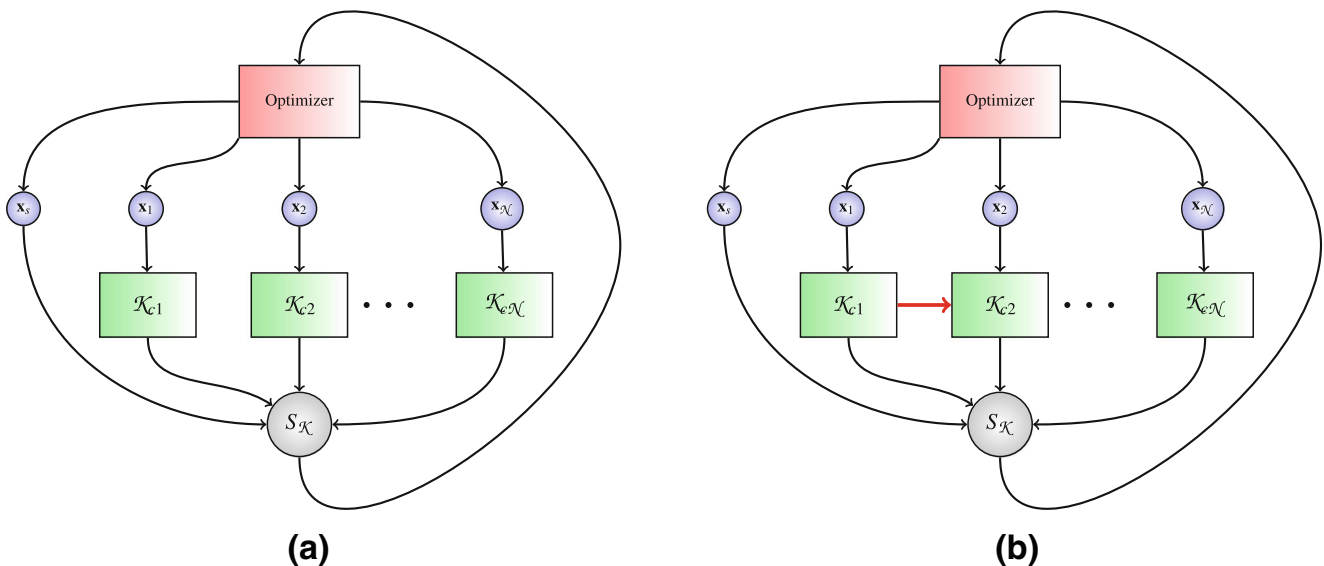
$$\text{Corr}[Y(\mathbf{x}_i), Y(\mathbf{x}_j)] = \exp\left(-\sum_{q=1}^k \theta_q (x_{iq} - x_{jq})^2\right) \tag{3}$$

where  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are any two locations in the domain and  $k$  is the number of dimensions and  $\theta_q, \mu, \sigma^2$  are the parameters to be estimated via Maximum Likelihood Estimation (MLE).

The Kriging prediction  $\hat{y}$  at a previously unsampled location then uses the MLE of the Kriging parameters. The Kriging prediction  $\hat{y}$  is approximated by

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \hat{\mathbf{r}}^T \hat{\mathbf{R}}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) \tag{4}$$

where  $\hat{\mu}$  is the estimated mean,  $\hat{\mathbf{R}}$  is the  $N \times N$  estimated correlation matrix between the  $N$  samples,  $\hat{\mathbf{r}}$  is the vector of estimated correlations between the observed data and the new prediction and  $\mathbf{y}$  is the observed response.



**Fig. 1** a shows the process of optimization of the system based on metamodels of the independent components is shown. In b, a type of problem in which dependence exists between the underlying components of the system is illustrated

An important aspect of Kriging is that a potential error in the interpolation can be estimated. The MSE estimate in the Kriging prediction is approximated by the following equation, which ignores the noise created when the unknown Kriging parameters are estimated,

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[ 1 - \hat{\mathbf{r}}^T \hat{\mathbf{R}}^{-1} \hat{\mathbf{r}} + \frac{1 - \mathbf{1}^T \hat{\mathbf{R}}^{-1} \hat{\mathbf{r}}}{\mathbf{1}^T \hat{\mathbf{R}}^{-1} \mathbf{1}} \right], \quad (5)$$

where  $\hat{\sigma}^2$  is the MLE of  $\sigma^2$ . The error increases as the distance between the new point and the old points increases (if the new point coincides with an old point, then the error is exactly zero).

### 3.1.2 Efficient global optimization

The combination of the Kriging prediction  $\hat{y}(\mathbf{x})$  and MSE  $s^2(\mathbf{x})$  can be used to adaptively sample the design domain in order to quickly estimate the global minimum (Jones et al. 1998). One of the most effective Kriging based adaptive sampling criteria is that of Expected Improvement (EI). EI is formulated by exploiting the fact that Kriging assumes a Gaussian Process, where the marginal distribution of  $\hat{y}(\mathbf{x})$  is normal. An improvement can be made over the current observed minimum  $y_{\min}$  at  $\mathbf{x}$  if some part of  $Y(\mathbf{x})$  is below  $y_{\min}$ . The Probability of Improvement (PI) over  $y_{\min}$  may be expressed as,

$$P[I(\mathbf{x})] = \Phi \left( \frac{y_{\min} - \hat{y}}{s} \right), \quad (6)$$

where  $s$  is the square root of  $s^2$  in (5) and  $\Phi(\cdot)$  is the normal cumulative distribution function. A new sampling location that gives the highest probability of improvement over  $y_{\min}$  can be found by maximizing (6). The use of this infill sampling criterion will be discussed in more detail in Section 3.1.3, where it finds its use for dealing with constrained problems.

An alternative method of infill sampling involves finding the expectation of the amount of improvement over  $y_{\min}$ . The EI can be calculated by taking the expectation  $E[I(\mathbf{x})] = E[\max(y_{\min} - Y, 0)]$ . A computationally cheap analytical expression for EI can be found in terms of the normal cumulative distribution function  $\Phi(\cdot)$  and the normal probability density function  $\phi(\cdot)$  (Schonlau and Welch 1996). Under the condition that  $\hat{y} < y_{\min}$ , the EI over the minimum observed response at any location  $\mathbf{x}$  is

$$E[I(\mathbf{x})] = (y_{\min} - \hat{y}) \Phi \left( \frac{y_{\min} - \hat{y}}{s} \right) + s \phi \left( \frac{y_{\min} - \hat{y}}{s} \right). \quad (7)$$

A new sampling location that gives the maximum *expected* improvement can be found by estimating the maximizer of (7).

The adaptive sampling strategy can be explained as follows. An initial metamodel  $\mathcal{K}_f$  is constructed using  $n$  samples chosen via Design of Experiments, e.g. Latin Hypercube sampling (LHS). Equation 7 is then maximized to estimate the new adaptive sampling location  $\mathbf{x}_{new}$ . The sampling location and response is added to the set of samples and responses. The metamodel is constructed again with the augmented set of samples and responses until either  $EI_{\max}$ , the maximum EI at the last iteration, becomes less than  $\epsilon_{EI}$  or the computational budget  $N_T$  is exhausted. The algorithm is referred to as Efficient Global Optimization (EGO) (Jones et al. 1998). This EGO algorithm forms the main inspiration for the present work.

### 3.1.3 Optimization of constrained problems

An infill sampling strategy based on Kriging for computationally expensive constrained problems was first suggested by Schonlau (1997). The method involved building metamodels of the objective and constraints and iteratively improving both sets of metamodels via a single adaptive sampling scheme. This infill sampling criterion was simply a product of probability of improvement in the objective and a probability of *feasibility* for the constraint.

The probability of feasibility measure operates in much the same way as the probability of improvement for the objective. Metamodels are built for the expensive to evaluate constraint(s). Let the Kriging prediction and MSE for such a constraint metamodel be given by  $\hat{h}(\mathbf{x})$  and  $s^{2,(h)}(\mathbf{x})$ . The metamodel uncertainty for each constraint metamodel is again described in terms of a normally distributed random variable  $H(\mathbf{x})$  with mean  $\hat{h}(\mathbf{x})$  and variance  $s^{2,(h)}(\mathbf{x})$ .

Let  $h_{\min}$  be the constraint limit, so  $\hat{h}(\mathbf{x}) < h_{\min}$  (standard notation uses  $h_{\min} = 0$ , so the constrained problem at  $\mathbf{x}$  remains feasible as long as the constraint is less than or equal to zero; for an example we refer to (30).

The probability of feasibility is given by the area of the distribution  $H(\mathbf{x})$  that is below the constraint limit  $h_{\min}$ ,

$$P[F(\mathbf{x}) < h_{\min}] = \Phi \left( \frac{h_{\min} - \hat{h}}{s^{(h)}} \right). \quad (8)$$

A possible choice for a new *feasible* sampling criterion for a constrained problem can then be the product of the probability of improvement of the objective and the probability of feasibility of constraints. For a problem with a single constraint this may be written as,

$$PI_F = P[I(\mathbf{x})]P[F(\mathbf{x}) < h_{\min}] \quad (9)$$

Alternatively, the EI in the objective can be multiplied by the product of the probability of feasibility of the constraints,

$$EI_F = E[I(\mathbf{x})]P[F(\mathbf{x}) < h_{\min}]. \quad (10)$$

### 3.2 System level EI criterion

The aim of this work is to optimize (2) by means of building and iteratively improving a set of component metamodels such that the optimum of the original system problem, (1), is estimated. An additional goal is to solve system problems involving constraints. Changes required in the iterative sampling strategy to accommodate this goal are discussed in Section 3.3.

The algorithm is initialized by building Kriging metamodels of all components based on a set of initial samples. In a software implementation, these could also be pre-built library metamodels. The MSE of the individual component metamodels, denoted as  $s_i^2(\mathbf{x})$ , follows from (5). Let the system level response be given by

$$\hat{y}_{\text{sys}} = S_{\mathcal{K}}(\hat{y}_1(\mathbf{x}_1), \hat{y}_2(\mathbf{x}_2), \dots, \hat{y}_{\mathcal{N}}(\mathbf{x}_{\mathcal{N}}), x_s), \quad (11)$$

where  $\hat{y}_i$  is the Kriging prediction of each component  $\mathcal{K}_{ci}$ .

In order to update the metamodels in regions of interest for system optimization, an adaptive sampling strategy at system level is required. To derive such a system infill sampling criterion, an error estimate is needed of the system level response, (11).

The EI criterion in Section 3.1 was found by assuming that the uncertainty in the predicted value  $\hat{y}(\mathbf{x})$  at a position  $\mathbf{x}$  can be described in terms of a normally distributed random variable  $Y(\mathbf{x})$  with mean given by the Kriging prediction  $\hat{y}(\mathbf{x})$  and variance given by the Kriging MSE  $s^2(\mathbf{x})$ . Following the same process, we assume that the uncertainty in the system level response  $\hat{y}_{\text{sys}}$  at a position  $\mathbf{x}$  can be expressed in terms of a random variable  $Y_{\text{sys}}(\mathbf{x})$  with mean  $\hat{y}_{\text{sys}}$  and variance  $s_{\text{sys}}^2$ . Furthermore, we retain the assumption that the uncertainty in the predicted value of each component  $\mathcal{K}_{ci}$  can be described as a normally distributed random variable  $Y_i(\mathbf{x})$  with mean given by the Kriging prediction  $\hat{y}_i$  and variance given by the Kriging MSE  $s_i^2$ .

The random variable describing system level uncertainty,  $Y_{\text{sys}}(\mathbf{x})$ , can basically be described in terms of the system operation on the component level random variables  $\mathbf{Y} = [Y_1, Y_2, \dots, Y_{\mathcal{N}}]$  and the deterministic system variable  $\mathbf{x}_s$ ,

$$Y_{\text{sys}}(\mathbf{x}) = S_{\mathcal{K}}(Y_1(\mathbf{x}_1), Y_2(\mathbf{x}_2), \dots, Y_{\mathcal{N}}(\mathbf{x}_{\mathcal{N}}), \mathbf{x}_s), \quad (12)$$

where the normal random variables  $Y_1(\mathbf{x}_1), Y_2(\mathbf{x}_2), \dots, Y_{\mathcal{N}}(\mathbf{x}_{\mathcal{N}})$  are treated as being independent. Theoretically, these variables may be assumed to be dependent, but in practice this dependence may be ignored (Kleijnen and Mehdad 2014).

A linear approximation of the right hand side of (12) can be derived by performing the Taylor series expansion of  $S_{\mathcal{K}}(\mathbf{Y}, \mathbf{x}_s)$  about the mean values  $\hat{y}_i$  of  $\mathbf{Y}$  and truncating

the series to include only the linear terms, i.e. the first two terms,

$$S_{\mathcal{K}}(\mathbf{Y}, \mathbf{x}_s) = S_{\mathcal{K}}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{\mathcal{N}}, \mathbf{x}_s) + \sum_{i=1}^{\mathcal{N}} (Y_i - \hat{y}_i) \frac{\partial S_{\mathcal{K}}}{\partial Y_i} \Big|_{\hat{y}_i}. \quad (13)$$

Since the above expansion is linear,  $Y_{\text{sys}}$  is a normal random variable (Taboga 2012). It can be shown (Ayyub 2003) that the first order mean of  $Y_{\text{sys}}$  is given by

$$E(Y_{\text{sys}}) = \hat{y}_{\text{sys}} = S_{\mathcal{K}}[\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{\mathcal{N}}, \mathbf{x}_s] \quad (14)$$

and the first order variance of  $Y_{\text{sys}}$  is given by

$$\text{Var}(Y_{\text{sys}}) = s_{\text{sys}}^2 = \sum_{i=1}^{\mathcal{N}} b_i^2 s_i^2, \quad \text{where } b_i = \frac{\partial S_{\mathcal{K}}}{\partial Y_i} \Big|_{\hat{y}_i}. \quad (15)$$

The derivatives  $b_i$  can be computed cheaply using finite difference in case of black-box system functions, or analytically otherwise. This is based on the assumption that the components are expensive to evaluate, but the system is cheap. Since  $Y_{\text{sys}}$  is normally distributed, an analytical expression similar to EI, (7), can be derived for the system level EI as well.

Let the optimum on the system response, (2), be denoted by  $d_{\mathcal{K}}$ . We can improve on the value  $d_{\mathcal{K}}$  if  $Y_{\text{sys}} < d_{\mathcal{K}}$ . The expectation of this improvement,  $I_{\text{sys}} = \max(d_{\mathcal{K}} - Y_{\text{sys}}, 0)$ , can be found by

$$E[I_{\text{sys}}(\mathbf{x})] = \int_{I_{\text{sys}}=0}^{I_{\text{sys}}=\infty} I_{\text{sys}} \frac{\exp\left(-\frac{t_{\text{sys}}^2}{2}\right)}{\sqrt{2\pi} s_{\text{sys}}} dt_{\text{sys}}, \quad (16)$$

where

$$t_{\text{sys}} = \frac{d_{\mathcal{K}} - I_{\text{sys}} - \hat{y}_{\text{sys}}}{s_{\text{sys}}}. \quad (17)$$

The standard normal probability density function is given by

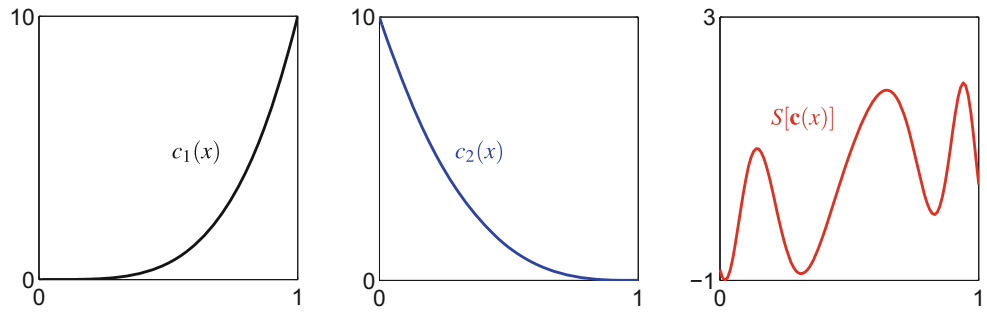
$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right). \quad (18)$$

Plugging in (18) into (16),  $EI_{\text{sys}}$  can be written as,

$$E[I_{\text{sys}}(\mathbf{x})] = (d_{\mathcal{K}} - \hat{y}_{\text{sys}}) \int_{t_{\text{sys}}=-\infty}^{t_{\text{sys}}=\frac{d_{\mathcal{K}}-\hat{y}_{\text{sys}}}{s_{\text{sys}}}} \phi(t_{\text{sys}}) dt_{\text{sys}} - s_{\text{sys}} \int_{t_{\text{sys}}=-\infty}^{t_{\text{sys}}=\frac{d_{\mathcal{K}}-\hat{y}_{\text{sys}}}{s_{\text{sys}}}} t_{\text{sys}} \phi(t_{\text{sys}}) dt_{\text{sys}}. \quad (19)$$

The first integral in (19) can be recognized as the normal cumulative distribution function  $\Phi\left(\frac{d_{\mathcal{K}}-\hat{y}_{\text{sys}}}{s_{\text{sys}}}\right)$ . The second integral can be solved by using the substitution  $z = \frac{-t_{\text{sys}}}{2}$ . Under the condition that  $\hat{y}_{\text{sys}} < d_{\mathcal{K}}$ , the final analytical

**Fig. 2** The components  $c_1(x)$  and  $c_2(x)$  and the system response  $S(\mathbf{c}(x))$  are plotted in the domain  $x \in [0, 1]$



expression for the infill sampling criterion at system level,  $EI_{sys}$ , is

$$E[I_{sys}(\mathbf{x})] = (d_{\mathcal{K}} - \hat{y}_{sys})\Phi\left(\frac{d_{\mathcal{K}} - \hat{y}_{sys}}{s_{sys}}\right) + s_{sys}\phi\left(\frac{d_{\mathcal{K}} - \hat{y}_{sys}}{s_{sys}}\right). \tag{20}$$

The location of the global maximum of (20) gives the next infill sampling location  $\mathbf{x}_{new}$ .

The algorithm is referred to as Bilevel Efficient Global Optimization (BEGO). We illustrate the main idea of the algorithm via the following illustrative example,

$$S(\mathbf{c}(x)) = \sin(c_1(x)) + \cos(c_2(x)),$$

where  $c_1(x) = 10x^4, c_2(x) = 10(1-x)^3, x \in [0, 1]$ . (21)

The system response is a function of two components that in turn are a function of the same variable  $x$ . Figure 2 shows the plot of  $S(\mathbf{c}(x))$ ,  $c_1(x)$  and  $c_2(x)$ . The system response has three local minima, two of which are very close in terms of their objective value.

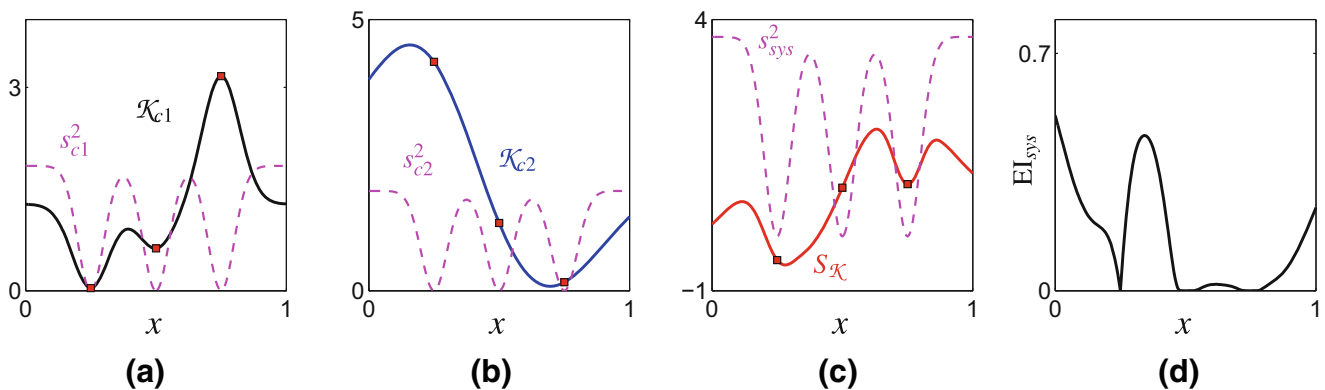
We construct Kriging metamodells of the components  $c_1(x)$  and  $c_2(x)$  based on a set of initial samples. The

true component and system responses are treated as black-box. The system response based on metamodells may be expressed as,

$$S_{\mathcal{K}}(\mathcal{K}_{\mathbf{c}}(x)) = \sin(\mathcal{K}_{c_1}(x)) + \cos(\mathcal{K}_{c_2}(x)), \quad x \in [0, 1]. \tag{22}$$

Figure 3 shows the Kriging metamodells,  $\mathcal{K}_{c_1}$  and  $\mathcal{K}_{c_2}$ , of the components  $c_1$  and  $c_2$  along with the system response  $S_{\mathcal{K}}$  and the system level EI,  $EI_{sys}$ . The metamodells have been initialized with three samples 0.25, 0.5 and 0.75. The MSE for  $\mathcal{K}_{c_1}$  and  $\mathcal{K}_{c_2}$  and the error estimate for  $S_{\mathcal{K}}$  are also shown in plots (a), (b) and (c), respectively. The error estimate for  $S_{\mathcal{K}}$  is constructed from the individual errors of  $\mathcal{K}_{c_1}$  and  $\mathcal{K}_{c_2}$ , by employing the linear transformation, (15), at each value of  $x$ . As expected, the error estimates are zero at the sample locations in plots (a), (b) and (c). Similarly, the plot of  $EI_{sys}$  shows that no improvement is expected at locations that have already been sampled.  $EI_{sys}$  is maximum at  $x = 0$  and this would be chosen as the new infill sampling location for this iteration of the algorithm.

Figure 4 shows a flowchart of the main steps involved in system optimization using Kriging based system EI. After constructing the component metamodells in Step 4, the new infill sampling location  $\mathbf{x}_{new}$  is estimated in Step 5. This involves a few substeps. Firstly, the optimum,  $d_{\mathcal{K}}$  on  $S_{\mathcal{K}}$  has



**Fig. 3** The Kriging metamodel of components  $c_1(x)$  and  $c_2(x)$  and the system response  $S_{\mathcal{K}}$  are plotted along with their respective error estimate in (a), (b) and (c). Plot (d) shows the system level expected improvement  $EI_{sys}$

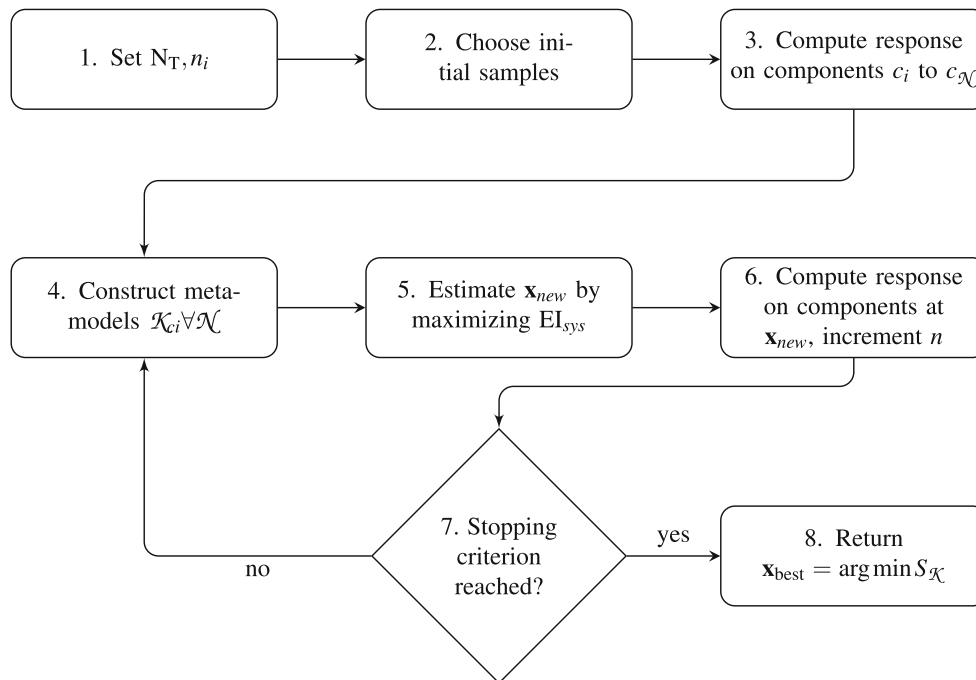


Fig. 4 Flowchart shows the steps involved in system optimization

to be estimated. Thereafter, the linear transformation of the component level MSE is performed to estimate the system level error,  $s_{sys}^2$ . The system error estimate  $s_{sys}^2$  is used along with the optimum  $d_{\mathcal{K}}$  to evaluate  $EI_{sys}$  in (20). The global maximum of (20) gives the new sampling location  $\mathbf{x}_{new}$ . In Step 6 the component response is evaluated at  $\mathbf{x}_{new}$ . If the stopping criterion has already been reached, then the argument of  $d_{\mathcal{K}}$  (namely, the minimizer of  $S_{\mathcal{K}}$ ) is returned as the final system optimum; otherwise, the algorithm returns to Step 4, and the loop is repeated until termination.

### 3.3 Constrained optimization of systems

If the system involves constraints then the sampling criterion must also be adapted to deal with these constraints. Similar to the formulation of system-level EI above, a probability of feasibility measure at system level can be derived to ensure that samples are added in areas of interest for constrained optimization instead of unconstrained optimization.

Let  $S_h$  represent the transformation of the component metamodels in order to get the constraint response  $\hat{h}_{sys}$ .

$$\hat{h}_{sys} = S_h(\hat{y}_1(\mathbf{x}_1), \hat{y}_2(\mathbf{x}_2), \dots, \hat{y}_N(\mathbf{x}_N), \mathbf{x}_s). \tag{23}$$

The error in the constraint response  $\hat{h}_{sys}$  can be derived in a manner that is completely analogous to the system error,  $s_{sys}$ , found for the objective. The system level variance for the constraint is denoted by  $s^{2,(h)}$ . The probability of

feasibility,  $PF_{sys}$ , for the constraint  $S_h$  can be expressed as,

$$P[F(\mathbf{x}) < h_{min}] = \Phi\left(\frac{h_{min} - \hat{h}_{sys}}{s^{(h)}}\right). \tag{24}$$

where  $h_{min}$  is the constraint limit that dictates whether the constraint is feasible or infeasible. The product of the system level EI,  $EI_{sys}$  and the system level probability of feasibility,  $PF_{sys}$ , gives a criterion for sampling a constrained problem. For a single constraint this criterion can simply be expressed as,

$$EI_{sf} = EI_{sys} \cdot PF_{sys}. \tag{25}$$

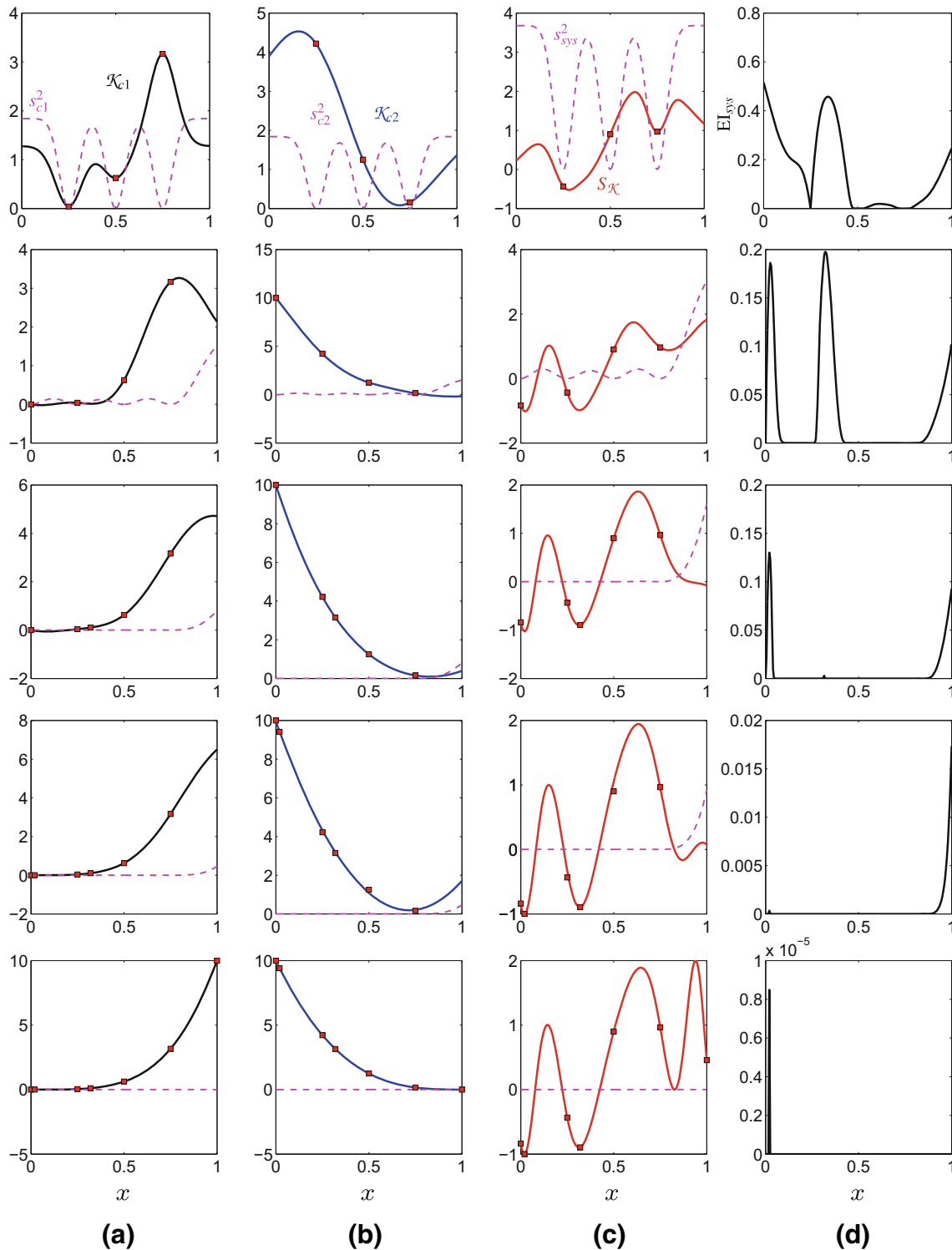
The maximizer of (25) is then sampled before constructing the component metamodels again. If there are multiple constraints, the product of individual probability of feasibility of all constraints will replace  $PF_{sys}$  in (25).

### 4 Numerical test problems

The algorithm is tested on several test problems to investigate its ability to estimate the optimum of the system accurately and consistently. The performance of the algorithm is compared against treating the entire system as a component and applying optimization using the EGO method (Jones et al. 1998). In the following discussion we

refer to the system decomposed into several components with the term DEC, (DEComposed), and the system treated as a single component with the term SAC, where SAC stands for System-As-Component. Similarly, we compare

the performance of the approach with an LHS with a fixed sample size based sampling scheme for approximate response construction and optimization of the SAC response.



**Fig. 5** The evolution of the method is shown as new sampling points are added at each iteration. The Kriging metamodel of components  $c_1(x)$  and  $c_2(x)$  and the system response  $S_{\mathcal{X}}$  are plotted along with their respective error estimate in (a), (b) and (c). Plot (d) shows the system EI,  $EI_{sys}$



The advantages of using an adaptive sampling scheme for the decomposed system problem are exhibited by comparing the effectiveness of the method with optimizing a system response for a constrained problem based on component metamodels built via one-shot LHS. A disadvantage of an adaptive sampling scheme such as BEGO is that the user is required to solve a multi-modal optimization problem at every iteration, (20), albeit on a cheap response. The optimization problem is multi-modal because  $EI = 0$  at each sample location that has already been simulated. Various methods for estimating the global optimum on this problem are mentioned in Kleijnen (2015), (p. 268). In this context, it should be noted that we operate under the assumption that the components are so computationally expensive that the internal computational complexity of BEGO is negligible in comparison. If the computational complexity of the underlying components is of the same order as the internal algorithm computation cost then the user may be better served to employ one-shot LHS.

#### 4.1 Algorithm evolution example

Before performing an in-depth analysis, we illustrate the evolution of the algorithm on the one-variable system that consisted of two non-linear components. The problem was introduced in Section 3, (21). The plot of the reference components and system response was given in Fig. 2.

Figure 5 shows five iterations of the algorithm, after it has been initialized with three samples at 0.25, 0.5 and 0.75 within the domain  $x \in [0, 1]$ . The component metamodels and the system response along with their respective error estimates are given in plot (a), (b) and (c) for each iteration. Plot (d) shows the system  $EI$  found at each iteration. Each subsequent iteration updates the component metamodel with  $\mathbf{x}_{new}$ , the location of the maximum value for  $EI_{sys}$  attained in the previous iteration. By the 5th iteration the maximum  $EI_{sys}$  falls to an insignificantly small value and the global optimum of the problem,  $S = -1$ , is found at  $x = 0.0196$ . The error estimates for the components and system response also drop fast. At the 5th iteration the component metamodels and the system response  $S_K$  match the reference in Fig. 2 and the respective error estimates are also very low.

The DEC and SAC algorithms were applied 20 times on the illustrative system problem  $S_1$ . Both methods were initialized with two random sampling locations for each replication. The average number of iterations needed by DEC to converge to the optimum was 6.6 while the corresponding number of iterations for SAC was 14.5. In part, this disparity is caused by the difference in nonlinearity of the system and component responses, which require varying amounts of samples to approximate appropriately.

To further illustrate this aspect, additional test problems are discussed below.

#### 4.2 Unconstrained test problems

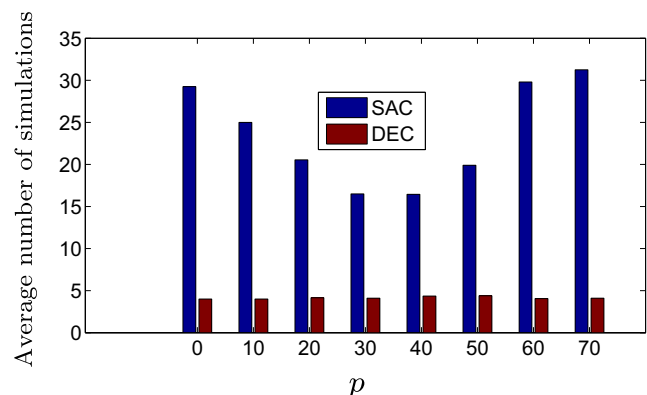
The algorithm is tested on a set of one-dimensional systems based on the following numerical problem,

$$S_p[\mathbf{c}(x)] = \sin(c_1(x)) + \cos(c_2(x)),$$

$$c_1(x) = px, \quad c_2(x) = (75 - p)x, \quad x \in [0, 1], \quad (26)$$

where  $p$  changes from  $p = 0$  to  $p = 70$  in steps of 10. This results in 8 different system problems with different values of the frequency  $p$ . Figure 6 shows the number of iterations needed to find the optimum for each of these 8 problems by SAC and DEC. Each problem was initialized with two random sampling locations for both SAC and DEC. In order to ensure that the result is not skewed by fortuitously hitting the optimum, both algorithms are only terminated when the optimal solution is repeated over 2 consecutive iterations. The bar chart shows the average performance over 20 runs for both methods. DEC found the optimum for each problem using less than 5 iterations. On the other hand SAC needed more than 15 iterations in all cases. For all values of  $p$ , the functions had multiple local optima that were relatively close to each other, so the problem was relatively difficult to optimize even in one variable.

A question that arises from the clear difference between the performance of DEC and SAC in Fig. 6 is why DEC does so much better than SAC on all problems. The fact is that the metamodels that DEC has to build are very simple linear functions that require only relatively few function calls to fit. On the other hand SAC has to construct the metamodel of the system response which is highly nonlinear. Therefore SAC requires many more iterations to fit the response well enough to estimate the optimum correctly.



**Fig. 6** Comparison of the average number of simulations, based on 20 runs, needed by DEC and SAC to converge to the optimum of the problem. The horizontal axis is the value of the problem parameter  $p$

To test this argument, we apply the following problem on SAC and DEC,

$$\begin{aligned}
 S_q[\mathbf{c}(x)] &= c_1(x) + c_2(x), \\
 c_1(x) &= (5(x - 0.5))^2 + 20 \sin(qx), \\
 c_2(x) &= (5(x - 0.5))^2 - 20 \sin(qx), \quad x \in [0, 1], \quad (27)
 \end{aligned}$$

where the frequency  $q$  changes from  $q = 10$  to  $q = 100$  in steps of 10. This results in ten problems with differing frequency of the components. The system transformation is linear while the component responses are highly nonlinear. It should be noted however that if SAC is used to optimize (27) then the response is always linear and always the same, no matter what value is chosen for  $q$ . This is due to the fact that the sinusoidal parts of the components cancel each other at system level.

Both SAC and DEC are initialized with two random samples. Again, in order to ensure that the algorithm is not terminated when the algorithm accidentally finds the optimum, both algorithms are stopped when the optimum is repeated over two consecutive iterations. Based on an average of 20 runs SAC requires only 5 total function calls to locate the optimum. Figure 7 shows the performance of DEC for different values of  $q$ . For all values of  $q$ , DEC requires more simulations than SAC to converge to the optimum. But interestingly the difference between the performance of SAC and DEC is not as dramatic in Fig. 7 as was the case for the problems shown in Fig. 6. This example, nevertheless, demonstrates that in certain specific cases, such as the one described above, a system problem with independent components should not be decomposed. The choice of whether to decompose the problem or not, depends on the relative nonlinearity of the system and components, respectively.

The algorithm is next tested on the three-hump camel function, which is an established test problem

(Surjanovic and Bingham 2013) for unconstrained optimization in two dimensions,

$$\begin{aligned}
 f_{camel} &= 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2, \\
 x_i &\in [-5, 5] \forall i \in \mathcal{N}. \quad (28)
 \end{aligned}$$

In the decomposed system level problem, we assume that there are three expensive components and the rest of the function is cheap to compute. The problem is decomposed and written in terms of the system and components as,

$$\begin{aligned}
 S_{camel}[\mathbf{c}(\mathbf{x})] &= 2c_1(x_1) - 1.05c_1^2(x_1) + \frac{c_1^3(x_1)}{6} \\
 &\quad + c_2(x_1, x_2) + c_3(x_2), \\
 c_1(x_1) &= x_1^2, \quad c_2(x_1, x_2) = x_1x_2, \quad c_3(x_2) = x_2^2, \\
 x_i &\in [-5, 5] \forall i \in \mathcal{N}. \quad (29)
 \end{aligned}$$

The proposed algorithm is applied 100 times on the system in (29). The component models are constructed using 4 initial samples based on LHS. The number of expensive simulations is limited to 12. Therefore the algorithm can run for 8 iterations.

The mean performance of the 100 runs of the system level approach is compared against applying optimization on a Kriging metamodel of the original function in (28) which is constructed using 12 expensive simulations chosen via LHS. The Kriging metamodel is also constructed 100 times based on 100 different combinations of LHS samples and the mean performance is analyzed.

Table 1 shows the comparison of the two approaches along with the reference global optimum of the function. The mean and standard deviation of the objective value at the global optimum location found by both methods are shown. The objective values shown have been generated on the original function, (28), as a post-processing step. The mean optimum found by adaptive system sampling approach for the decomposed problem is closer to the reference solution than the one found by LHS. In addition, the standard deviation for the system sampling scheme is also much lower compared to the one found by using LHS. This indicates that performing adaptive system sampling of the decomposed problem in (29) is more efficient than

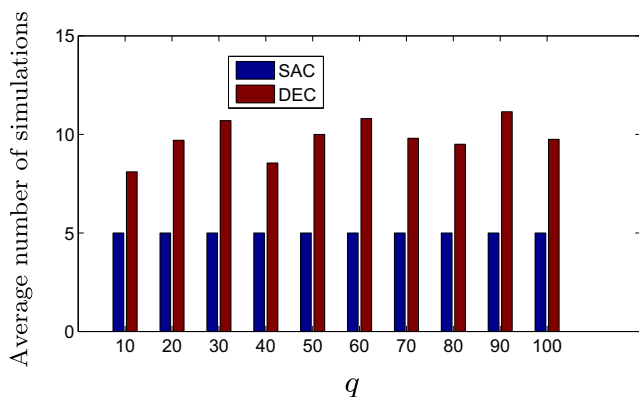


Fig. 7 The average number of simulations, based on 20 runs, needed by DEC and SAC to converge to the optimum of the problem. The horizontal axis is the value of problem parameter  $q$

Table 1 Comparison of adaptive system optimization of decomposed Camel problem, (29), versus application of LHS on the original Camel function, (28). The mean and standard deviation of the global optimum found by each method is shown based on 100 runs

Optimum	Mean	Standard deviation	Expensive function evaluations
Reference	0	—	—
Adaptive system sampling	0.00	0.001	12
Latin Hypercube sampling	1.02	0.015	12

applying LHS on the original problem in (28). However, a comparison of the performance of LHS and the adaptive sampling scheme on a decomposed problem would also be interesting. In the following subsection we show this comparison for the more widely applicable case of a constrained problem.

### 4.3 Constrained test problem

The algorithm is now applied on a constrained benchmark problem. The objective is a modified version of the Rastrigin function (Mühlenbein et al. 1991). The constrained problem is given below,

$$\begin{aligned}
 f(\mathbf{x}) &= 0.01 \left( 100 + \sum_{i=1}^{10} [x_i^2 - 10 \cos(0.2\pi x_i)] \right)^2 \\
 \text{s.t. } & 35 - \sum_{i=1}^4 x_i^2 \leq 0, \\
 & x_i \in [-5.12, 5.12] \forall i \in \mathcal{N}. \quad (30)
 \end{aligned}$$

The problem consists of 10 variables. We choose to decompose the problem into a system with four expensive components. Based on this decomposition, the system can be written as,

$$\begin{aligned}
 S[\mathbf{c}(\mathbf{x})] &= 0.01 (100 + c_1 - 10c_2 + c_3 - 10c_4)^2 \\
 & \text{s.t. } 35 - c_1 \leq 0, \\
 c_1 &= \sum_{i=1}^4 x_i^2, \quad c_2 = \sum_{i=1}^4 \cos(0.2\pi x_i), \\
 c_3 &= \sum_{i=5}^{10} x_i^2, \quad c_4 = \sum_{i=5}^{10} \cos(0.2\pi x_i), \\
 & x_i \in [-5.12, 5.12] \forall i \in \mathcal{N}. \quad (31)
 \end{aligned}$$

The system non-linearly transforms the response of the four non-linear components. The components  $c_1$  and  $c_2$  are a function of four variables which are shared across both the components. On the other hand,  $c_3$  and  $c_4$  are a function of six different variables, which are also shared across these two components. The component  $c_1$  makes an appearance in both the objective and the constraint. The metamodel of  $c_1$  can therefore be used for both the system transformation,  $S_{\mathcal{K}}$ , of the objective as well as in the system transformation,  $S_h$ , of the constraint.

The decomposed problem in (31) is used to compare the performance of BEGO with an LHS based strategy. For this problem, metamodels are made for the component response instead of the system response for the LHS based approach as well. Both methods are run 20 times on the proposed problem, (31), and the mean performance is compared.

In the case of BEGO, the component metamodels are initialized with a different number of samples based on the

number of dimensions of the component. The four dimensional components  $c_1$  and  $c_2$  are initialized with 20 samples, while the six dimensional components  $c_3$  and  $c_4$  are initialized with 30 samples. The algorithm is allowed to run for 80 iterations. This means that at termination, 100 samples would have been added to  $\mathcal{K}_{c_1}$  and  $\mathcal{K}_{c_2}$ , while  $\mathcal{K}_{c_3}$  and  $\mathcal{K}_{c_4}$  would be based on 110 samples.

The LHS based approach is therefore given a total budget of 100 samples for  $c_1$  and  $c_2$  and 110 samples for  $c_3$  and  $c_4$ . The system response  $S_{\mathcal{K}}$  based on the LHS based metamodels is optimized and the average result of the 20 runs is analyzed.

We also compare the performance for the case in which both LHS and BEGO are given 25 less expensive simulations. This means that BEGO is terminated after 55 iterations. On the other hand, for LHS  $\mathcal{K}_{c_1}$  and  $\mathcal{K}_{c_2}$  are constructed based on 75 samples and  $\mathcal{K}_{c_3}$  and  $\mathcal{K}_{c_4}$  are built using 85 samples.

Figure 8 shows the average global optimum found at each iteration of BEGO, based on 20 runs. The objective is evaluated, as a post-processing step, on the reference system response based on the optimal location found at each iteration of BEGO. The reference global optimum is also found on the reference system. The errorbars indicate the standard deviation around the optimum. After about 60 iterations of BEGO, the algorithm has converged to a value close to the reference global optimum and the standard deviation is also relatively low.

Table 2 shows the comparison of the statistics of the global optimum found by LHS and BEGO. Once again the objective values given here have been found by evaluating the optima found by each method on the reference system response as a post-processing step. The results indicate that BEGO performs better than LHS in terms of mean closeness of the optimum to the reference solution, given the

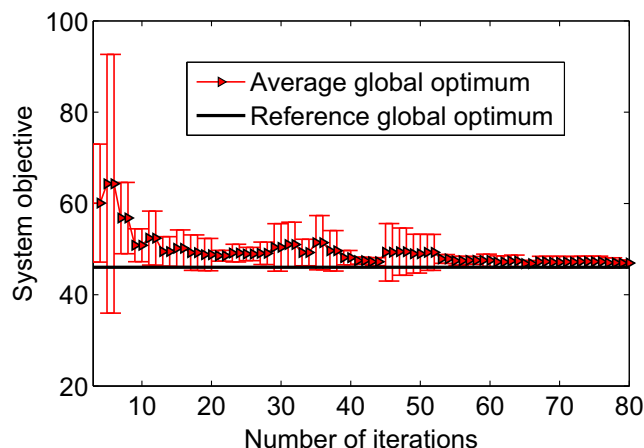


Fig. 8 The mean and standard deviation of the global optimum found on each iteration of BEGO based on the 20 runs for the test problem defined in (31)

**Table 2** Comparison of the mean and standard deviation of the global optimum found by BEGO and LHS based on 20 runs for small and large number of observations

Optimum	Reference	BEGO	LHS	Evaluations $c_1, c_2$	Evaluations $c_3, c_4$
Mean	46.03	47.43	48.89	75	85
Standard deviation	—	0.97	2.12	75	85
Constraint violations	—	0	0	75	85
Mean	46.03	46.93	47.70	100	110
Standard deviation	—	0.66	1.34	100	110
Constraint violations	—	0	2	100	110

same number of component evaluations. The standard deviation of BEGO is also lower than for LHS. Interestingly, for LHS with a higher number of samples, 2 out of the 20 optimal locations found result in a constraint violation. On the other hand, the optimal locations of BEGO do not violate the constraint for any of the 20 runs. Comparing the result of the higher number of evaluations (second set of rows in Table 2) with the result for the lower number of evaluations, we note that convergence is steadily taking place for both approaches. The mean and standard deviation for BEGO based on the lower number of evaluations are both better than the corresponding numbers for LHS with a higher number of evaluations.

Using the constrained and unconstrained problems, we have evaluated the performance of BEGO in different settings. The method converges well on the problems and also performs better than a space-filling based strategy. The primary advantage of the adaptive technique lies in reducing the computational budget needed to find the global optimum. Since each extra component simulation significantly increases the time and computational costs of optimization, any amount of reduction in these costs brings significant efficiency improvement. We now test the algorithm in a

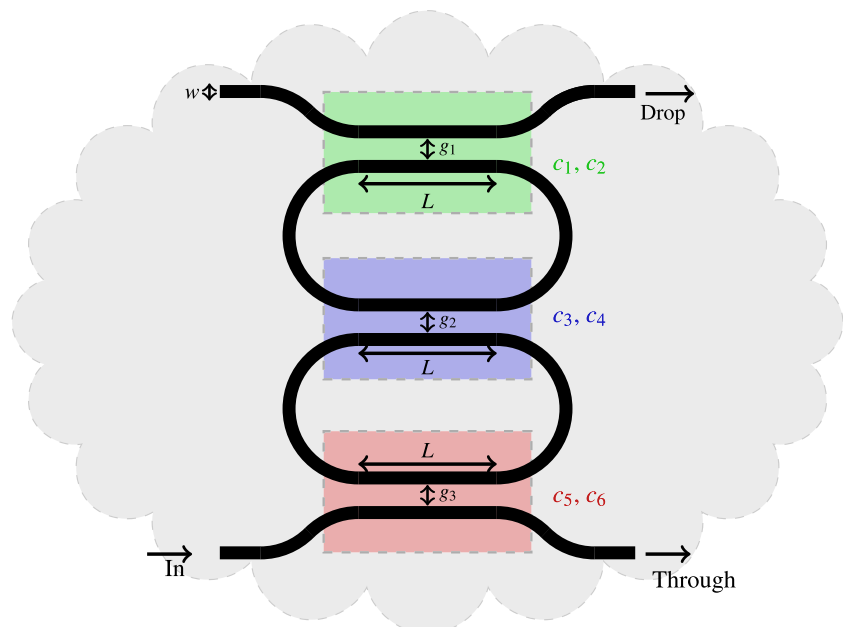
more practical setting by applying it on an engineering problem.

## 5 Engineering problem: Optimization of an optical filter

### 5.1 Background of the problem

We perform system optimization of an integrated photonic serial ring resonator that behaves as an optical filter. Light is confined in a high refractive index SiN core layer which is buried inside a relatively lower refractive index SiO<sub>2</sub> cladding layer. The light is guided inside these optical integrated circuits via the principle of total internal reflection. Figure 9 shows the top-view schematic of the system. When light at a certain wavelength is launched into the waveguide (black paths lines) at the ‘In’ port, it partially couples into the adjacent waveguide. This coupled light travels through the first ring section and partially couples again into the second ring section. A portion of this coupled light is dropped at the ‘Drop’ port. The rest of the light exits via the ‘Through’ port.

**Fig. 9** Top-view schematic of a second order serial optical ring resonator. The cloud encompasses the complete system. The colored boxes, on the other hand, represent the six expensive to evaluate components



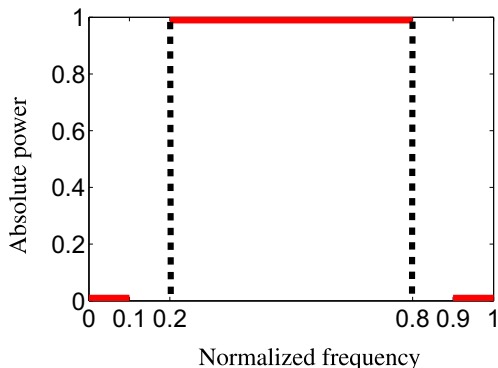
The amount of coupling can be varied by changing the gap and the length of the coupling sections (area enclosed by the colored boxes in Fig. 9). This change in coupling gives rise to a change in the optical response at the ‘Drop’ and ‘Through’ port. Different optical filter responses can therefore be generated by varying the coupling ratio of each coupler. The objective in this study is to achieve the band-pass filter response given in Fig. 10 at the ‘Through’ port.

In order to compute the power coupling ratio of the coupler, two expensive electromagnetic simulations have to be performed. The first simulation computes the power coupling ratio,  $P_{L0}$ , when the coupling length is zero. The second simulation computes the beat length,  $L_{\pi}$ , which is defined as the coupling length needed to completely couple light from the first waveguide into the second waveguide and back into the first waveguide. Since the exchange of light between adjacent waveguides follows a sinusoidal curve, the two parameters,  $P_{L0}$  and  $L_{\pi}$ , are enough to compute the coupled power at any coupling length given a certain gap (Syms and Cozens 1992). Figure 11 shows the power coupling ratio as a function of the coupling length based on a certain simulated value of  $P_{L0}$  and  $L_{\pi}$  for a given gap.

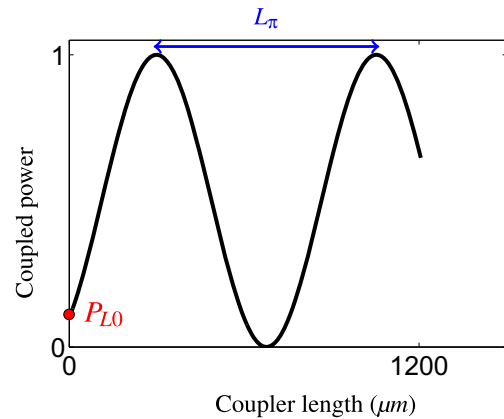
Once the power coupling ratio of each coupler is known, the transfer function at the ‘Through’ port can be cheaply computed as a function of the normalized frequency using scattering matrix analysis (Ahmed et al. 2011). Let  $H(n_f)$  represent the transfer function for the normalized frequency  $n_f \in [0 1]$ . We can then define a system objective for the desired bandpass spectral response as,

$$\min_{\mathbf{x} \in \mathbb{X}} \frac{1-b}{2} \|\bar{H}_{stop1}\|_p + b \left[ 1 - \|\bar{H}_{pass}\|_p \right] + \frac{1-b}{2} \|\bar{H}_{stop2}\|_p, \tag{32}$$

where  $\bar{H}_{stop1}$ ,  $\bar{H}_{pass}$  and  $\bar{H}_{stop2}$  represent the vector of responses for  $n_f \in [0 0.1]$ ,  $n_f \in [0.2 0.8]$  and  $n_f \in [0.8 0.9]$



**Fig. 10** Desired band-pass filter spectral response (red line) of the serial ring resonator. No preference is specified in the intervals [0.1, 0.2] and [0.8, 0.9]



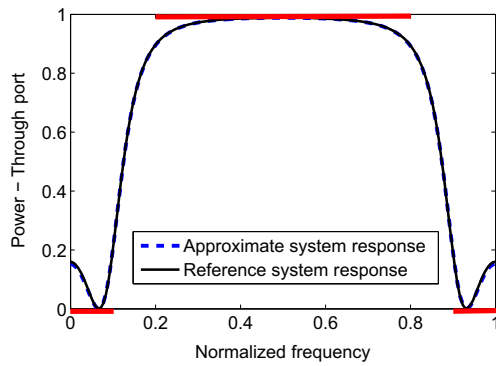
**Fig. 11** Power coupling ratio as a function of the coupler length for a certain coupler gap

[0.9 1], respectively. The five design variables of the problem, depicted in Fig. 9, are  $\mathbf{x} \in [w g_1 g_2 g_3 L]$ , where  $w \in [1 1.15]\mu m$ ,  $L \in [0 2400]\mu m$  and all the gaps are in the range  $[1 1.3]\mu m$ . If  $p$  is large then the objective basically involves minimization of the weighed sum of the maximum value in the two stop bands and the minimum value in the pass-band. The  $p$ -norm is used instead of the maximum and minimum value in order to ensure that the objective remains continuously differentiable. The use of the  $p$ -norm instead of the minimum and maximum values enables faster convergence for the algorithm since the objective is relatively easier to model. The weights are based on the value of  $b$ . In this work, we choose  $p = 20$  and  $b = 0.6$ . The value of  $p$  was chosen such that there is a balance between smoothness of the objective and the need to closely approximate the maximum value.

### 5.2 System optimization

The system response,  $S$ , can be modeled in terms of two expensive components that are repeated three times in the second order resonator since each coupler requires two expensive simulations and there are three couplers in the system. Figure 9 shows the six components  $c_1$  to  $c_6$  as well as the five design variables of the problem  $[w g_1 g_2 g_3 L]$ . We treat the responses  $P_{L0}$  and  $L_{\pi}$ , generated from each coupling section, as the components of the problem since this suits the mathematical structure of this problem. The expensive components are only a function of the width and the respective gaps. The components  $c_1$ ,  $c_3$  and  $c_5$  give the power coupling ratio  $P_{L0}$  for each coupler while the beat length  $L_{\pi}$  for each coupler is found by components  $c_2$ ,  $c_4$  and  $c_6$ .

A commercial electromagnetic solver, PhoeniX Software, is used to compute  $P_{L0}$  and  $L_{\pi}$ . Both simulations require approximately 10 minutes. Initial Kriging metamod-els for  $c_1$  and  $c_2$  are built based on  $n_d = 9$  samples for  $w$



**Fig. 12** Plot shows the approximate system response and the reference system response at the global optimum found by BEGO

and  $g_1$  chosen via LHS. We assign  $g_2$  and  $g_3$  the same initial sample values as  $g_1$ . Since the design domain for all the components is the same we can essentially use  $\mathcal{K}_{c1}$  as the approximate response for  $P_{L0}$  for all three components  $c_1$ ,  $c_3$  and  $c_5$ . Similarly  $\mathcal{K}_{c2}$  can be used to give the beat length  $L_\pi$  for components  $c_2$ ,  $c_4$  and  $c_6$ .

Once the component metamodels that predict the value for  $P_{L0}$  and  $L_\pi$  for each coupler have been constructed, the following operations to evaluate the system objective are performed at system level. In order to evaluate the objective to be optimized, (32), the power coupling ratio of each coupler is needed. This operation is performed on system level since the value of  $P_{L0}$  and  $L_\pi$  can cheaply be predicted by the Kriging metamodels for each coupler. The computation of the transfer function via scattering matrix analysis involves simple operations on small matrices and is therefore also performed on system level. Once the transfer function is known, it has to be plugged into (32) to find the system objective.

A total computational budget of 21 expensive simulations for evaluating  $P_{L0}$  and  $L_\pi$  is reserved in addition to the 9 initial simulations. At each iteration of the system algorithm,  $P_{L0}$  and  $L_\pi$  is simulated three times for the three different combinations  $[w \ g_1]$ ,  $[w \ g_2]$  and  $[w \ g_3]$  of the location for infill sampling suggested by the algorithm. The total number of simulations available translates to 7 iterations of the system optimization algorithm since  $7 \times 3 = 21$  simulations. The three new  $P_{L0}$  responses at each iteration are all added to the Kriging metamodel  $\mathcal{K}_{c1}$ . On the other hand,  $\mathcal{K}_{c2}$  is augmented with the set of three new beat length responses  $L_\pi$  at each iteration.

**Table 3** The location of the global optimum is given along with the objective on the approximate system and the reference system

	w	g1	g2	g3	L	$S_{\mathcal{K}}$	S
Optimum	1.0699	1.085	1.100	1.084	1940.2	0.0159	0.0160

**Table 4** The response of the individual components at the global optimum are given for the metamodels and the reference simulator

Components	$c_1(P_{L0})$	$c_3(P_{L0})$	$c_5(P_{L0})$	$c_2(L_\pi)$	$c_4(L_\pi)$	$c_6(L_\pi)$
Reference	0.1203	0.1126	0.1203	903.45	937.83	903.45
Kriging	0.1203	0.1130	0.1203	903.62	938.00	903.62

Figure 12 shows the optimal system response found after 7 iterations of BEGO. The approximate system response  $S_{\mathcal{K}}$  is plotted along with the system response obtained when the optimal result is fed into the expensive component simulators. It can be seen from the figure that the system response based on the component metamodels closely resembles the actual system response at the global optimum. The optimum found by BEGO seems to perform quite well in the pass band region, i.e. almost all the light in the normalized frequency region [0.2 0.8] is being passed through. The amount of power in the stop bands [0 0.1], [0.9 1] is still not very low. This is to be expected since we are optimizing a second order filter, i.e. there are only two rings in the serial resonator structure. Increasing the filter order will further improve the performance.

Table 3 shows the location at which the global optimum of the serial ring resonator is found. Although we do not impose the restriction that  $g_3 = g_1$ , the optimal result produces a resonator with symmetric gaps. The numerical objective value for  $S$  and  $S_{\mathcal{K}}$  confirm the result shown in Fig. 12. Table 4 gives the response of the individual components at the global optimum. As expected, the Kriging metamodel responses for  $P_{L0}$  and  $L_\pi$  for each directional coupler section is close to the reference component response.

## 6 Conclusion

In this work, we have developed an efficient strategy for global optimization of systems involving independent components. The approach, referred to as BEGO, targets systems involving expensive to evaluate components that do not interact with each other. Kriging metamodels were employed to construct the responses of the expensive components. A novel system level infill sampling strategy was derived in order to sample the components in regions of interest for system optimization. A linear Taylor series approximation of the system transformation of the Kriging components was performed in order to obtain an analytical expression for system level EI. The infill sampling criterion was modified for constrained system optimization by deriving a system level probability of feasibility for each constraint.

The system level optimization approach was first compared with treating the problem as a single component and

applying EGO (Jones et al. 1998) for obtaining the optimum. Similarly, the effectiveness of the technique was compared with building a response surface of the system via LHS and globally optimizing the cheap response. Both comparisons exhibited that, in general, there is clear efficiency improvement when optimization is performed on the decomposed problem. However, if the component responses are highly non-linear and the system response is linear then decomposition may not always result in efficiency improvement.

We demonstrated the advantages of using an adaptive sampling scheme for the decomposed system problem by comparing the effectiveness of the strategy with performing optimization of a system response based on component metamodels constructed using LHS.

The two approaches were applied on a modified and constrained version of the Rastrigin function. Based on an average of 20 runs, BEGO converged comparatively faster to the optimum of the problem as opposed to the LHS based optimization strategy.

In addition, BEGO was applied on an engineering system involving an optical filter based on an integrated photonic serial ring resonator. The engineering problem consisted of a system with six uncoupled components. It was shown that BEGO was able to determine the global optimum of the system problem using only a limited number of expensive component simulations.

The algorithm is especially relevant for problems in which many multiple components are part of the system. In such a situation the same metamodel can be used to construct the response for the multiple components. Another advantage of employing the bilevel framework is that the number of dimensions for each component metamodel is often much lower than the total dimension size at system level. This enables the approximate system response to converge fast to the actual response with each passing component level simulation. Since the component response is treated as a black-box, the method is applicable to any hierarchical system with low dimensional components.

In this work, we have addressed optimization of systems involving independent components. A future aim is to extend the algorithm to the case where the components are dependent. We expect that the extension of the algorithm to systems of dependent components will have to draw upon methods developed within the field of Multidisciplinary Design Optimization (Martins and Lambe 2013).

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Ahmed OS, Swillam MA, Bakr MH, Li X (2011) Efficient Design Optimization of Ring Resonator-Based Optical Filters. *J Lightwave Technol* 29(18):2812–2817. doi:10.1109/JLT.2011.2162818
- Ayyub BM (2003) Risk Analysis in Engineering and Economics. Chapman & Hall/CRC, Boca Raton, FL
- Barton RR (1997) Design of experiments for fitting subsystem metamodels Proceedings of the 29th conference on Winter simulation, IEEE Computer Society, pp 303–310
- Forrester A, Sobester A, Keane A (2008) Engineering design via surrogate modelling: a practical guide. John Wiley & Sons, UK
- Hu W, Li M, Azarm S, Almansoori A (2011) Multi-Objective Robust Optimization Under Interval Uncertainty Using Online Approximation and Constraint Cuts. *J Mech Des* 133(6):061,002. doi:10.1115/1.4003918
- Jones DR, Schonlau M, Welch WJ (1998) Efficient Global Optimization of Expensive Black-Box Functions. *J Glob Optim* 13(4):455–492. doi:10.1023/A:1008306431147
- Kleijnen JP (2015) Design and analysis of simulation experiments. Springer
- Kleijnen JP, Mehdad E (2014) Multivariate versus univariate kriging metamodels for multi-response simulation models. *Eur J Oper Res* 236(2):573–582. doi:10.1016/j.ejor.2014.02.001
- Koch PN, Mavris D, Mistree F (2000) Partitioned, Multilevel Response Surfaces for Modeling Complex Systems. *AIAA J* 38(5):875–881. doi:10.2514/2.1042
- Kokkolaras M, Mourelatos ZP, Papalambros PY (2004) Design optimization of hierarchically decomposed multilevel systems under uncertainty ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, p. 613–624
- Li G, Azarm S, Farhang-Mehr A, Diaz AR (2006) Approximation of multiresponse deterministic engineering simulations: a dependent metamodeling approach. *Struct Multidiscip Optim* 31(4):260–269. doi:10.1007/s00158-005-0574-5
- Lifante G (2003) Integrated photonics: fundamentals. J. Wiley. doi:10.1002/0470861401
- Martins JRR, Lambe AB (2013) Multidisciplinary Design Optimization: A Survey of Architectures. *AIAA J* 51(9):2049–2075. doi:10.2514/1.J051895
- Mühlenbein H, Schomisch M, Born J (1991) The parallel genetic algorithm as function optimizer. *Parallel Comput* 17(6):619–632. doi:10.1016/S0167-8191(05)80052-3
- Sacks J, Welch W, Mitchell TJ, Wynn HP (1989) Design and Analysis of Computer Experiments. *Stat Sci* 4(4):409–435
- Schonlau M (1997) Computer experiments and global optimization. PhD thesis, University of Waterloo
- Schonlau M, Welch W (1996) Global optimization with nonparametric function fitting Proceedings of the Section on Physical and Engineering Sciences, American Statistical Association, pp 183–186
- Surjanovic S, Bingham D (2013) Virtual Library of Simulation Experiments: Test Functions and Datasets. <http://www.sfu.ca/ssurjano>
- Syms R, Cozens J (1992) Optical guided waves and devices. McGraw-Hill
- Taboga M (2012) Lectures on probability theory and mathematical statistics., 2nd edn. CreateSpace Independent Publishing Platform
- Viana FAC, Simpson TW, Balabanov V, Toropov V (2014) Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come? *AIAA J* 52(4):670–690. doi:10.2514/1.J052375
- Xiu R, Zhang X, Liu Y, Huang HZ (2013) Metamodeling uncertainty quantification in multi-level engineering system design Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE), 2013 International Conference on, pp 449–454. doi:10.1109/QR2MSE.2013.6625621