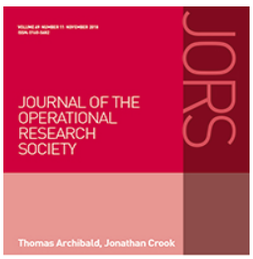

Journal of the Operational Research Society

ISSN: 0160-5682 (Print) 1476-9360 (Online) Journal homepage: https://www.tandfonline.com/loi/tjor20

Efficient global optimisation for black-box simulation via sequential intrinsic Kriging

Ehsan Mehdad & Jack P. C. Kleijnen

To cite this article: Ehsan Mehdad & Jack P. C. Kleijnen (2018) Efficient global optimisation for black-box simulation via sequential intrinsic Kriging, Journal of the Operational Research Society, 69:11, 1725-1737, DOI: 10.1080/01605682.2017.1409154

To link to this article: https://doi.org/10.1080/01605682.2017.1409154

Published online: 16 Jan 2018.

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY, 2018
VOL. 69, NO. 11, 1725–1737
https://doi.org/10.1080/01605682.2017.1409154

Efficient global optimisation for black-box simulation via sequential intrinsic Kriging

Ehsan Mehdad and Jack P. C. Kleijnen

Tilburg School of Economics and Management, Tilburg University, Tilburg, Netherlands

ABSTRACT

Efficient global optimisation (EGO) is a popular method that searches sequentially for the global optimum of a simulated system. EGO treats the simulation model as a black-box, and balances local and global searches. In deterministic simulation, classic EGO uses ordinary Kriging (OK), which is a special case of universal Kriging (UK). In our EGO variant we use intrinsic Kriging (IK), which does not need to estimate the parameters that quantify the trend in UK. In random simulation, classic EGO uses stochastic Kriging (SK), but we replace SK by stochastic IK (SIK). Moreover, in random simulation, EGO needs to select the number of replications per simulated input combination, accounting for the heteroscedastic variances of the simulation outputs. A popular method uses optimal computer budget allocation (OCBA), which allocates the available total number of replications to simulated combinations. We replace OCBA by a new allocation algorithm. We perform several numerical experiments with deterministic simulations and random simulations. These experiments suggest that (1) in deterministic simulations, EGO with IK outperforms classic EGO; (2) in random simulations, EGO with SIK and our allocation rule does not perform significantly better than EGO with SK and OCBA.

ARTICLE HISTORY
Received 18 December 2015
Accepted 16 November 2017

KEYWORDS
Global optimisation; Gaussian process; Kriging; intrinsic Kriging; metamodel

1. Introduction

Optimisation methods for black-box simulations – either deterministic or random – have many applications, as our references will demonstrate. Black-box simulation means that the input/output (I/O) function is treated as an implicit mathematical function defined by the simulation model (computer code). In many engineering applications of computational fluid dynamics, the computation of the output (response) of a single input combination is time-consuming or "computationally expensive". In most operational research (OR) applications, however, a single simulation run is computationally inexpensive, but there are extremely many input combinations; e.g., a single-server queueing model may have one input (namely, the traffic rate) that is continuous, so we can distinguish infinitely many input values but in finite time, we can simulate only a fraction of these values. In all these situations, it is common to use metamodels, which are also called emulators or surrogates. A popular method for the optimisation of deterministic simulation is *efficient global optimisation* (EGO), which uses Kriging metamodels; see the seminal article Jones, Schonlau, and Welch (1998). EGO has been adapted for random simulation with either homoscedastic noise variances (see Huang, Allen, Notz, and Zeng (2006)) or heteroscedastic noise variances (see Picheny, Ginsbourger, Richet, and Caplin (2013) and Quan, Yin, Ng, and Lee (2013)). EGO is the topic of much recent research; see Mehdad and Kleijnen (2015a), Sun, Hong, and Hu (2014), Salemi, Nelson, and Staum (2014), and the many references in Kleijnen (2015, p. 267). Besides EGO, there are other search strategies; see Hoad, Monks, and O'Brien (2015).

Our first contribution in this paper is the use of *intrinsic Kriging* (IK) metamodels for the optimisation of deterministic simulation and random simulation. To the best of our knowledge, IK has not yet been applied in EGO. The basic idea of IK is to remove the trend from I/O data by linear filtration of data. Consequently, IK does not require the second-order stationary condition and it may provide a more accurate fit than Kriging; see Mehdad and Kleijnen (2015b) and Mehdad (2015).

More specifically, for deterministic simulation, we develop EGO with IK for the metamodel. For random simulation, we develop stochastic IK (SIK) combined with the two-stage sequential algorithm developed by Quan et al. (2013). The latter algorithm accounts for heteroscedastic noise variances and balances two source of noise; namely, spatial uncertainty due to the metamodel and random variability caused by the simulation. The latter noise is independent from one replication to another; i.e., we suppose that the streams of pseudo-random numbers do not overlap. Moreover we assume that different input combinations do not use common (pseudo)random numbers (CRN), so these input combinations give independent outputs.

Our second contribution concerns Quan et al.'s

CONTACT Ehsan Mehdad emehdad@gmail.com

two-stage algorithm. We replace the *optimal computing budget allocation* (OCBA) in the allocation stage of the algorithm by an allocation rule that we build on IK. Our rule allocates the additional replications to the sampled points such that it minimises the integrated mean squared prediction error (IMSPE). Our rule revises the allocation scheme that was originally developed in Ankenman, Nelson, and Staum (2010). Note that automatic selection of the number of replications is also investigated in Hoad, Robinson, and Davies (2010).

In our numerical experiments we use test functions of different dimensionality, to study the differences between (1) EGO variants in deterministic simulation; (2) two-stage algorithm variants in random simulation. Our major conclusion will be that in most experiments (1) for deterministic simulations, EGO with IK outperform EGO formulated in Jones et al. (1998); (2) for stochastic (random) simulations, the performance difference between Quan's et al. algorithm and our new algorithm is not significant.

We organise the rest of this paper as follows. Section 2 summarises classic Kriging. Section 3 explains IK. Section 4 summarises classic EGO, the two-stage algorithm, and our algorithm. Section 5 presents numerical experiments. Section 6 summarises our conclusions.

2. Kriging

In this section we summarise *universal Kriging* (UK), following Cressie (1991, pp. 151–182). UK assumes

$$Y(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} + M(\mathbf{x}) \text{ with } \mathbf{x} \in \mathbb{R}^d \qquad (1)$$

where $Y(\mathbf{x})$ is a random process at the point (or input combination) $\mathbf{x}$, $\mathbf{f}(\mathbf{x})$ is a vector of $p+1$ known regression functions or *trend*, $\boldsymbol{\beta}$ is a vector of $p+1$ parameters, and $M(\mathbf{x})$ is a stationary Gaussian process (GP) with zero mean and covariance function Σ_M.

This Σ_M must be specified such that it indeed makes $M(\mathbf{x})$ in (1) a stationary GP; i.e., Σ_M is a function of the *distance* between the points $\mathbf{x}_i$ and $\mathbf{x}_{i'}$ with $i, i' = 0, 1, \ldots, m$ where the subscript 0 denotes a new point and m denotes the number of old points. *Separable anisotropic* covariance functions use the distances along the d axes $h_{i;i';g} = |x_{i;g} - x_{i';g}|$ $(g = 1, \ldots, d)$. The most popular choice for such a function is the so-called *Gaussian* covariance function:

$$\text{cov}(\mathbf{x}_i, \mathbf{x}_{i'}) = \tau^2 \prod_{g=1}^{d} \exp\left(-\theta_g h_{i;i';g}^2\right) \text{ with } \quad \theta_g > 0 \qquad (2)$$

where τ^2 is the variance of $M(\mathbf{x})$.

Let $\mathbf{Y} = \left(Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_m)\right)^\top$ denote the vector with the m values of the metamodel in (1) at the m old points. Kriging predicts Y at a (new or old) point $\mathbf{x}_0$ *linearly* from the old I/O data $(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ is the $d \times m$ matrix with m points $(\mathbf{x}_i = (x_{g;i})$ $(i = 1, \ldots, m;\ g = 1, \ldots, d)$:

$$\hat{Y}(\mathbf{x}_0) = \boldsymbol{\lambda}^\top \mathbf{Y} \text{such that } \boldsymbol{\lambda}^\top \mathbf{F} = \mathbf{f}(\mathbf{x}_0)^\top \qquad (3)$$

where $\mathbf{F}$ is the $m \times (p+1)$ matrix with element (i, j) being $f_j(\mathbf{x}_i)$, $\mathbf{f}(\mathbf{x}_0) = \left(f_0(\mathbf{x}_0) \ldots, f_p(\mathbf{x}_0)\right)^\top$, and the condition for $\boldsymbol{\lambda}$ guarantees that $\hat{Y}(\mathbf{x}_0)$ is an *unbiased* predictor. The *optimal* linear unbiased predictor minimises the *mean squared prediction error* (MSPE), defined as:

$$\text{MSPE}(\hat{Y}(\mathbf{x}_0)) = \mathbb{E}(\hat{Y}(\mathbf{x}_0) - Y(\mathbf{x}_0))^2.$$

Cressie (1991, pp. 151–157) shows how to use Lagrangian multipliers to solve this constrained minimisation problem, which gives the *optimal* weights and the predictor:

$$\begin{aligned} \boldsymbol{\lambda}^\top &= \left(\boldsymbol{\Sigma}_M(\mathbf{x}_0, \cdot) + \mathbf{F}\left(\mathbf{F}^\top \boldsymbol{\Sigma}_M^{-1} \mathbf{F}\right)^{-1} \right. \\ &\quad \left. \times \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \boldsymbol{\Sigma}_M^{-1} \boldsymbol{\Sigma}_M(\mathbf{x}_0, \cdot)\right)\right)^\top \boldsymbol{\Sigma}_M^{-1} \\ \hat{Y}(\mathbf{x}_0) &= \boldsymbol{\lambda}^\top \mathbf{Y} \end{aligned} \qquad (4)$$

with $\Sigma_M(\mathbf{x}_0, \cdot) = \left(\Sigma_M(\mathbf{x}_0, \mathbf{x}_1), \ldots, \Sigma_M(\mathbf{x}_0, \mathbf{x}_m)\right)^\top$ denoting the m-dimensional vector with covariances between the outputs of the one new point and the m old points, and $\boldsymbol{\Sigma}_M$ denoting the $m \times m$ matrix with the covariances between the old outputs, so the element (i, i') is $\Sigma_M(\mathbf{x}_i, \mathbf{x}_{i'})$. The resulting minimal MSPE or predictor variance is:

$$\begin{aligned} \text{MSPE}(\hat{Y}(\mathbf{x}_0)) = \tau^2 &- \boldsymbol{\Sigma}_M(\mathbf{x}_0, \cdot)^\top \boldsymbol{\Sigma}_M^{-1} \boldsymbol{\Sigma}_M(\mathbf{x}_0, \cdot) \\ &+ \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \boldsymbol{\Sigma}_M^{-1} \boldsymbol{\Sigma}_M(\mathbf{x}_0, \cdot)\right)^\top \\ &\times \left(\mathbf{F}^\top \boldsymbol{\Sigma}_M \mathbf{F}\right)^{-1} \\ &\times \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \boldsymbol{\Sigma}_M^{-1} \Sigma_M(\mathbf{x}_0, \cdot)\right). \end{aligned} \qquad (5)$$

This Kriging is an exact interpolator; i.e., for the old points, (4) gives a predictor that equals the observed output. Important for EGO is the property that for the old points the predictor variance (5) reduces to zero.

The Kriging metamodel defined in (1) can be extended to incorporate the so-called *internal* noise in random simulation; see the following references sorted in historical order: Opsomer, Ruppert, Wand, Holst, and Hossjer (1999), Ginsbourger (2009), Ankenman et al. (2010), and Yin, Ng, and Ng (2011). The resulting *stochastic Kriging* (SK) metamodel at replication r of the random simulation output at $\mathbf{x}$ is:

$$Z_r(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} + M(\mathbf{x}) + \varepsilon_r(\mathbf{x}) \text{ with } \mathbf{x} \in \mathbb{R}^d \qquad (6)$$

where $\varepsilon_r(\mathbf{x})$ denotes the internal noise in replication r of point $\mathbf{x}$. We assume that $\varepsilon_r(\mathbf{x})$ has a Gaussian distribution with mean zero and variance $V(\mathbf{x})$ and that it is independent of $M(\mathbf{x})$.

We can then derive the SK predictor and its MSPE analogously to the derivation for UK in (4) and (5) except that we replace $\mathbf{\Sigma} = \mathbf{\Sigma}_M + \mathbf{\Sigma}_{\bar{\varepsilon}}$ by $\mathbf{\Sigma}_M$ where $\mathbf{\Sigma}_{\bar{\varepsilon}}$ is a diagonal matrix (no CRN) with the variances of the internal noise $V(\mathbf{x}_i)/n_i$ (where n_i is the number of replication at $\mathbf{x}_i$) on the main diagonal and $\mathbf{\Sigma}_M$ still denotes the covariance matrix of Kriging without internal noise. We also replace $\mathbf{Y}$ in (4) and (5) by the sample mean $\overline{\mathbf{Z}} = (\overline{Z}(\mathbf{x}_1), \ldots, \overline{Z}(\mathbf{x}_m))^\top$.

3. Intrinsic Kriging

In the previous section, we defined UK in (1). In UK with a *known* drift β, we assume $M(\mathbf{x})$ is a second-order stationary function with the covariance function $C(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(M(\mathbf{x}) - \beta)(M(\mathbf{x}') - \beta)]$. This covariance function is bounded; i.e., $\left|C(\mathbf{x}, \mathbf{x}')\right| \leq \tau^2$ where τ^2 is the variance of $M(\mathbf{x})$. In case we do not know β, we cannot compute $M(\cdot) - \beta$ in $C(\mathbf{x}, \mathbf{x}')$. We therefore use another tool called the *structure function* or *variogram* $\gamma_M(\mathbf{x}, \mathbf{x}') = \mathrm{Var}\left[M(\mathbf{x}) - M(\mathbf{x}')\right]/2$. The variogram shows the dissimilarity between $M(\mathbf{x})$ and $M(\mathbf{x}')$ for any pair of observed points $\mathbf{x}$ and $\mathbf{x}'$. The variogram is bounded for a second-order stationary function, but can increases to infinity for a non-stationary function. A second-order stationary function implies $\gamma_M(\mathbf{x}, \mathbf{x}') = \tau^2 - C_M(\mathbf{x}, \mathbf{x}')$. Unfortunately the drift affects γ_M. More specifically, Chilès and Delfiner (2012, p. 125) shows that even with an *optimal* estimator $\hat{\beta}$ of the drift, the variogram of the *estimated* residual is biased downward:

$$\gamma_M(\mathbf{x}, \mathbf{x}') = \gamma_Y(\mathbf{x}, \mathbf{x}') - \frac{1}{2}\mathrm{Var}\left(\hat{\beta}(\mathbf{x}) - \hat{\beta}(\mathbf{x}')\right).$$

How should we then remove the drift from the I/O data? We start with the simple case of a *constant drift*. To remove such a drift, we use *intrinsic random functions* (IRFs). These IRFs have *increments* $Y(\mathbf{x}) - Y(\mathbf{x}_0)$ that remove a constant drift; moreover, these increments are second-order stationary. This stationarity enables us to calculate the variance for a linear combination of increments, which is essential to compute a linear predictor with minimum variance. But how is a linear combination of increments related to a linear predictor (linear combination of metamodel values at old observations)? We can define a linear combination of increments $\sum_i \lambda_i \left(Y(\mathbf{x}) - Y(\mathbf{x}_0)\right)$ only in terms of $Y(\mathbf{x})$. Any linear combination $\sum_i \lambda_i Y(\mathbf{x})$ that satisfies $\sum_i \lambda_i = 0$ is equivalent to $\sum_i \lambda_i \left(Y(\mathbf{x}) - Y(\mathbf{x}_0)\right)$. Such linear combinations with weights adding up to zero are called 'allowable linear combinations'. We can show that the covariance of two allowable linear combinations $\sum_i \lambda_i Y(\mathbf{x}_i)$ and $\sum_j \mu_j Y(\mathbf{x}_j)$ is

$$\mathrm{Cov}\left(\sum_i \lambda_i Y(\mathbf{x}_i), \sum_j \mu_j Y(\mathbf{x}_j)\right) = -\sum_i \sum_j \lambda_i \mu_j \gamma_Y(\mathbf{x}_j - \mathbf{x}_i).$$

So allowable linear combinations of IRFs remove a constant drift and have finite variance. Hence, we can define a new linear predictor with minimum variance that does not depend on a constant trend. It is also possible to construct similar linear predictors that do not depend on a more general trend (e.g., a linear, or a quadratic trend; see $\beta(\mathbf{x})$ in (1)) in the I/O data.

Now we discuss a wider class of non-stationary functions called *intrinsic random function of order k* (IRF-k). An IRF-k has increments of order k that are second-order stationary, and also eliminate a polynomial drift of order k. To formalise such an IRF-k, we follow Mehdad and Kleijnen (2015b). So, we rewrite (1) as:

$$\mathcal{Y} = \mathbf{F}\boldsymbol{\beta} + \mathcal{M} \tag{7}$$

where $\mathbf{F} = (F_{ij} = f_j(\mathbf{x}_i))$ $(i = 1, \ldots, m,\ j = 1, \ldots, p+1)$, $\mathcal{Y} = \left(\mathcal{Y}(\mathbf{x}_1), \ldots, \mathcal{Y}(\mathbf{x}_m)\right)^\top$, and $\mathcal{M} = \left(\mathcal{M}(\mathbf{x}_1), \ldots, \mathcal{M}(\mathbf{x}_m)\right)^\top$. We do not assume $\mathcal{M}$ is second-order stationary. Let $\boldsymbol{\lambda}$ be an $m \times 1$ vector such that $\boldsymbol{\lambda}^\top \mathbf{F} = \mathbf{0}^\top$ where $\mathbf{0}$ is a $(p+1)$-dimensional vector of zeros. Together, $\boldsymbol{\lambda}$ and (7) give

$$\boldsymbol{\lambda}^\top \mathcal{Y} = \boldsymbol{\lambda}^\top \mathcal{M}.$$

Consequently, the second-order properties of $\boldsymbol{\lambda}^\top \mathcal{Y}$ depend on $\boldsymbol{\lambda}^\top \mathcal{M}$ and *not on the regression function* $\mathbf{F}\boldsymbol{\beta}$ in (7).

To generalise the model in (1), we need a stochastic process for which $\boldsymbol{\lambda}^\top \mathcal{M}$ is second-order stationary; such processes are called *intrinsically* stationary processes. We assume that $f_j(\mathbf{x})$ $(j = 1, \ldots, p+1)$ are mixed monomials $x_1^{i_1} \cdots x_d^{i_d}$ with $\mathbf{x} = (x_1, \ldots, x_d)^\top$ and non-negative integers $i_1, \ldots, i_d$ such that $i_1 + \ldots + i_d \leq k$ with k a given non-negative integer. An *intrinsic random function of order k* (IRF-k) is a random process $\mathcal{Y}$ for which $\sum_{i=1}^m \lambda_i \mathcal{Y}(\mathbf{x}_i)$ with $\mathbf{x}_i \in \mathbb{R}^d$ is second-order stationary. This $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)^\top$ is called the *generalised-increment* vector which must satisfy the following conditions:

$$\left(f_j(\mathbf{x}_1), \ldots, f_j(\mathbf{x}_m)\right)\boldsymbol{\lambda} = 0 \ (j = 1, \ldots, p+1).$$

Let Λ_k be the class of all generalised-increments of order k. For a one-dimensional function, $\boldsymbol{\lambda} \in \Lambda_k$ with

$k \in \{0, 1, 2\}$ must satisfy the following conditions:

$$\sum_{i=1}^{m} \lambda_i = 0 \quad \text{if} \quad k = 0 \text{ (constant drift);}$$
$$\sum_{i=1}^{m} \lambda_i = 0, \sum_{i=1}^{m} \lambda_i x_i = 0 \quad \text{if} \quad k = 1 \text{ (linear drift);}$$
$$\sum_{i=1}^{m} \lambda_i = 0, \sum_{i=1}^{m} \lambda_i x_i = 0, \sum_{i=1}^{m} \lambda_i x_i^2 = 0$$
if $k = 2$ (quadratic drift).

Obviously, an IRF-k is also an IRF-$(k+1)$, so $\Lambda_{k+1} \subset \Lambda_k$.

Note: The increments of a second-order stationary process are also second-order stationary, so UK with a second-order stationarity assumption (which is popular in applications) is an IRF itself.

We know that an IRF itself does not have a finite variance or its variance may depend on $\mathbf{x}$. However, we can calculate the variance of a linear combination of the ordinary increments of an IRF (equivalent to an allowable linear combination of order 0) in terms of a *variogram*. The covariance structure for any two allowable linear combinations of order k is called a *generalised covariance function* (GCF) K. To derive properties of this K, we follow Mehdad and Kleijnen (2015b). Obviously K is symmetric; so $K(\mathbf{x}_i, \mathbf{x}_{i'}) = K(\mathbf{x}_{i'}, \mathbf{x}_i)$. Moreover, K must be *conditionally* positive definite so

$$\forall \boldsymbol{\lambda}, \quad \text{var}(\boldsymbol{\lambda}^\top \mathcal{Y}) = \sum_{i=1}^{m} \sum_{i'=1}^{m} \lambda_i \lambda_{i'} K(\mathbf{x}_i - \mathbf{x}_{i'}) \geq 0,$$
$$\text{such that} \quad \left(f_j(\mathbf{x}_1), \ldots, f_j(\mathbf{x}_m)\right) \boldsymbol{\lambda} = 0$$

where the condition must hold for $j = 1, \ldots, p+1$. This condition makes $\boldsymbol{\lambda}$ a generalised increment vector of order k. Mehdad and Kleijnen (2015b) discusses different models for GCFs and suggests

$$K(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \prod_{g=1}^{d} \left(\theta_{0;g} + \theta_{1;g} \int_0^1 \frac{(x_g - u_g)_+^{k_g} (x'_g - u_g)_+^{k_g}}{(k_g!)^2} du_g\right) \quad (8)$$

where $\boldsymbol{\theta} = (\theta_{0;1}, \theta_{1;1}, \theta_{0;2}, \ldots, \theta_{0;d}, \theta_{1;d}) \geq 0$. The function in (8) accepts different k for the d different input dimensions, so we have a vector of the *orders* $\mathbf{k} = (k_1, \ldots, k_d)^\top$.

We introduce IK based on an IRF-k. Let $\mathcal{M}(\mathbf{x})$ be an IRF-k with mean zero and GCF K. Then the IK metamodel is:

$$\mathcal{Y}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} + \mathcal{M}(\mathbf{x}). \quad (9)$$

The IK metamodel predicts $\mathcal{Y}$ at a new point $\mathbf{x}_0$ using a linear combination of observed data $\mathcal{Y}(\mathbf{x})$. Using proper constraints on the weights, we guarantee that the prediction error $\hat{\mathcal{Y}}(\mathbf{x}_0) - \mathcal{Y}(\mathbf{x}_0)$ is an allowable linear combination of order k.

Actually, Cressie (1991, pp. 299–306) derives a linear predictor for the IRF-k metamodel defined in (9) with GCF K. We have the old outputs $\boldsymbol{\mathcal{Y}} = \left(\mathcal{Y}(\mathbf{x}_1), \ldots, \mathcal{Y}(\mathbf{x}_m)\right)^\top$ with the generalised covariance matrix $\mathbf{K}$. The optimal linear prediction of $\mathcal{Y}(\mathbf{x}_0)$ at a new location $\mathbf{x}_0$ follows from minimising the mean squared prediction error (MSPE) of the linear predictor:

$$\min_{\boldsymbol{\lambda}} E\left(\hat{\mathcal{Y}}(\mathbf{x}_0) - \mathcal{Y}(\mathbf{x}_0)\right)^2 \text{ such that } \hat{\mathcal{Y}}(\mathbf{x}_0) = \boldsymbol{\lambda}^\top \boldsymbol{\mathcal{Y}}. \quad (10)$$

IK should meet the additional condition

$$\boldsymbol{\lambda}^\top \mathbf{F} = \left(f_0(\mathbf{x}_0), \ldots, f_p(\mathbf{x}_0)\right), \quad (11)$$

which guarantees that the coefficients of the prediction error $\lambda_1 \mathcal{Y}(\mathbf{x}_1) + \ldots + \lambda_m \mathcal{Y}(\mathbf{x}_m) + \lambda_0 \mathcal{Y}(\mathbf{x}_0)$ create a generalised-increment vector $\boldsymbol{\lambda}_{m+1}^\top = \left(\boldsymbol{\lambda}^\top, \lambda_0\right)$ with $\lambda_0 = -1$. This gives the variance of the IK predictor, denoted by σ_{IK}^2:

$$\sigma_{\text{IK}}^2 = \text{var}(\boldsymbol{\lambda}_{m+1}^\top \boldsymbol{\mathcal{Y}}) = \sum_{i=0}^{m} \sum_{i'=0}^{m} \lambda_i \lambda_{i'} K(\mathbf{x}_i, \mathbf{x}_{i'}). \quad (12)$$

Temporarily we assume that K is *known*, so the optimal linear predictor is obtained through minimisation of (12) subject to (11). Hence, the IK predictor is given by (10) with

$$\boldsymbol{\lambda}^\top = \left(\mathbf{K}(\mathbf{x}_0, \cdot) + \mathbf{F}\left(\mathbf{F}^\top \mathbf{K}^{-1} \mathbf{F}\right)^{-1} \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot)\right)\right)^\top \mathbf{K}^{-1} \quad (13)$$

where $\mathbf{K}(\mathbf{x}_0, \cdot) = \left(\mathbf{K}(\mathbf{x}_0, \mathbf{x}_1), \ldots, \mathbf{K}(\mathbf{x}_0, \mathbf{x}_m)\right)^\top$ and $\mathbf{K}$ is an $m \times m$ matrix with (i, i') element $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_{i'})$. The resulting σ_{IK}^2 is given by:

$$\begin{aligned} \text{MSPE}(\hat{\mathcal{Y}}(\mathbf{x}_0)) = {} & \mathbf{K}(\mathbf{x}_0, \mathbf{x}_0) - \mathbf{K}(\mathbf{x}_0, \cdot)^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot) \\ & + \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot)\right)^\top \\ & \times \left(\mathbf{F}^\top \mathbf{K}^{-1} \mathbf{F}\right)^{-1} \\ & \times \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot)\right). \end{aligned} \quad (14)$$

In practice, however, K is *unknown* so we estimate the covariance function parameters (say) $\boldsymbol{\theta}$ in $K = K(\boldsymbol{\theta})$. For this estimation we use the *restricted maximum likelihood* (REML) estimator, which maximises the likelihood of *transformed* data that do not contain the unknown parameters of the drift. This transformation is close to the concept of *ordinary increments*. So we

assume $\mathcal{Y}$ is a *Gaussian* IRF-k. The REML estimator of $\boldsymbol{\theta}$ is then found through minimisation of the negative log-likelihood function

$$\ell(\boldsymbol{\theta}) = (m-q)/2\log(2\pi) - \frac{1}{2}\log\left|\mathbf{F}^\top\mathbf{F}\right| + \frac{1}{2}\log|\mathbf{K}(\boldsymbol{\theta})| + \frac{1}{2}\log\left|\mathbf{F}^\top\mathbf{K}(\boldsymbol{\theta})^{-1}\mathbf{F}\right| + \frac{1}{2}\mathcal{Y}^\top\Xi(\boldsymbol{\theta})\mathcal{Y} \quad (15)$$

where $q = \text{rank}(\mathbf{F})$ and $\Xi(\boldsymbol{\theta}) = \mathbf{K}(\boldsymbol{\theta})^{-1} - \mathbf{K}(\boldsymbol{\theta})^{-1}\mathbf{F}\left(\mathbf{F}^\top\mathbf{K}(\boldsymbol{\theta})^{-1}\mathbf{F}\right)^{-1}\mathbf{F}^\top\mathbf{K}(\boldsymbol{\theta})^{-1}$; we denote the resulting estimator by $\hat{\boldsymbol{\theta}}$. Finally, we replace $\mathbf{K}$ by $\mathbf{K}(\hat{\boldsymbol{\theta}})$ in (13) – to obtain $\widehat{\boldsymbol{\lambda}}$ – and in (14) – to obtain $\hat{\sigma}^2_{\text{IK}}$.

Note: We could require REML to estimate the optimal $\mathbf{k}^*$ (with integer elements) too, but this would make the optimisation even more difficult. In our methodology, the user should try several values for $\mathbf{k}$ and pick the one that gives the best fit; see Mehdad and Kleijnen (2015b). Developing a procedure for finding $\mathbf{k}^*$ without such user intervention is a topic for future research.

Mehdad and Kleijnen (2015b) also extends IK to account for random simulation output with noise variances that change across the input space. The methodology is similar to the extension of Kriging to stochastic Kriging. The *interpolating* property of IK does not make sense for random simulation, which has sampling variability or internal noise (caused by pseudorandom numbers within the simulation model) – besides, the external noise or spatial uncertainty created by the fitted metamodel.

The literature has already extended IK to internal noise with a *constant* variance, called the *nugget effect* in geostatistics or *jitter* in machine learning. For example, Cressie (1991, p. 305) briefly discusses IK with a nugget effect, replacing $K(\mathbf{h})$ by $K(\mathbf{h}) + c_0\delta(\mathbf{h})$ where $c_0 \geq 0$, $\delta(\mathbf{h}) = 0$ if $\mathbf{h} > 0$, and $\delta(\mathbf{h}) = 1$ if $\mathbf{h} = 0$. Mehdad and Kleijnen (2015b) introduces Stochastic IK (SIK) that extends IK defined in (9) incorporating internal noise with heteroscedastic variances. The SIK metamodel at replication r of the random output at $\mathbf{x}$ is:

$$\mathsf{Y}_r(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top\boldsymbol{\beta} + \mathsf{M}(\mathbf{x}) + \varepsilon_r(\mathbf{x}) \text{ with } \mathbf{x} \in \mathbb{R}^d \quad (16)$$

where $\varepsilon_1(\mathbf{x}), \varepsilon_2(\mathbf{x}), \ldots$ denotes the internal noise at point $\mathbf{x}$. We assume that the internal noise has a Gaussian distribution with mean zero and variance $\mathsf{V}(\mathbf{x})$ (so the internal variance is heteroscedastic) and is independent of the external noise $\mathsf{M}(\mathbf{x})$.

In stochastic simulation, the experimental design consists of pairs $(\mathbf{x}_i, n_i)$, $i = 1, \ldots, m$, where n_i denotes the number of replications at $\mathbf{x}_i$. These replications enable us to compute the classic unbiased estimators of the mean output and the internal variance:

$$\overline{\mathsf{Y}}(\mathbf{x}_i) = \frac{\sum_{r=1}^{n_i}\mathsf{Y}_{i;r}}{n_i} \text{ and } s^2(\mathbf{x}_i) = \frac{\sum_{r=1}^{n_i}\left(\mathsf{Y}_{i;r} - \overline{\mathsf{Y}}(\mathbf{x}_i)\right)^2}{n_i - 1}. \quad (17)$$

We rewrite (16) as:

$$\overline{\mathsf{Y}}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top\boldsymbol{\beta} + \mathsf{M}(\mathbf{x}) + \overline{\varepsilon}(\mathbf{x}) \text{ with } \mathbf{x} \in \mathbb{R}^d \quad (18)$$

where $\mathsf{M}(\mathbf{x})$ is an IRF-k. Note that SIK (and SK) do not interpolate $\overline{\mathsf{Y}}(\mathbf{x})$; such interpolation is not desirable, because $\overline{\mathsf{Y}}(\mathbf{x})$ is no more than an estimate of the true value of the expected simulation response (in deterministic simulation, IK at old I/O points does interpolate the observed simulation responses).

Because we assumed that $\mathsf{M}(\mathbf{x})$ and $\varepsilon(\mathbf{x})$ in (16) are independent, the SIK predictor and its MSPE can be derived similarly to the IK predictor and its MSPE in (10) and (14) – replacing $\mathbf{K}_\mathsf{M}$ by $\mathbf{K} = \mathbf{K}_\mathsf{M} + \mathbf{K}_{\overline{\varepsilon}}$ where $\mathbf{K}_{\overline{\varepsilon}}$ is a diagonal matrix (so we assume that no CRN are used in the random simulation) with the variances of the internal noise $\mathsf{V}(\mathbf{x}_i)/n_i$ on the main diagonal; $\mathbf{K}_\mathsf{M}$ still denotes the generalised covariance matrix of IK without internal noise. We also replace $\mathcal{Y}$ in (10) and (14) by $\overline{\mathbf{Y}} = \left(\overline{\mathsf{Y}}(\mathbf{x}_1),\ldots,\overline{\mathsf{Y}}(\mathbf{x}_m)\right)^\top$. So the SIK predictor is

$$\widehat{\mathsf{Y}}(\mathbf{x}_0) = \boldsymbol{\lambda}^\top\overline{\mathbf{Y}} \text{ where } \boldsymbol{\lambda}^\top = \left(\mathbf{K}_\mathsf{M}(\mathbf{x}_0,\cdot) + \mathbf{F}\left(\mathbf{F}^\top\mathbf{K}^{-1}\mathbf{F}\right)^{-1}\left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top\mathbf{K}^{-1}\mathbf{K}_\mathsf{M}(\mathbf{x}_0,\cdot)\right)\right)^\top\mathbf{K}^{-1} \quad (19)$$

and its MSPE is:

$$\text{MSPE}(\widehat{\mathsf{Y}}(\mathbf{x}_0)) = \mathbf{K}_\mathsf{M}(\mathbf{x}_0,\mathbf{x}_0) - \mathbf{K}_\mathsf{M}(\mathbf{x}_0,\cdot)^\top\mathbf{K}^{-1}\mathbf{K}_\mathsf{M}(\mathbf{x}_0,\cdot) + \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top\mathbf{K}^{-1}\mathbf{K}_\mathsf{M}(\mathbf{x}_0,\cdot)\right)^\top \times \left(\mathbf{F}^\top\mathbf{K}^{-1}\mathbf{F}\right)^{-1} \times \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top\mathbf{K}^{-1}\mathbf{K}_\mathsf{M}(\mathbf{x}_0,\cdot)\right). \quad (20)$$

We use REML to obtain the estimator $\widehat{\boldsymbol{\theta}}$, and replace $\mathbf{K}_\mathsf{M}$ by $\mathbf{K}_\mathsf{M}(\widehat{\boldsymbol{\theta}})$. We also need to estimate the internal noise V. We use the following IK metamodel for the internal noise:

$$\mathsf{V}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top\boldsymbol{\sigma} + \mathsf{Z}(\mathbf{x}) \quad (21)$$

where $\mathbf{f}(\mathbf{x})^\top\boldsymbol{\sigma}$ is the regression function and Z is an IRF-k independent of M. Because $\mathsf{V}(\mathbf{x})$ is not observable – even at old points $\mathbf{x}_i$ $(i = 1, \ldots, m)$ – we let $s^2(\mathbf{x}_i)$ defined in (17) replace $\mathsf{V}(\mathbf{x}_i)$. The IK metamodel in (21) implies that we assume the $s^2(\mathbf{x}_i)$ have no noise and $\widehat{\mathsf{V}}(\mathbf{x}_i) = s^2(\mathbf{x}_i)$. We replace $\mathbf{K}_{\overline{\varepsilon}}$ by $\widehat{\mathbf{K}}_{\overline{\varepsilon}} = \left(\widehat{\mathsf{V}}(\mathbf{x}_1)/n_1, \ldots, \widehat{\mathsf{V}}(\mathbf{x}_m)/n_m\right)$. Finally, we replace $\mathbf{K} = \mathbf{K}_\mathsf{M} + \mathbf{K}_{\overline{\varepsilon}}$ by $\widehat{\mathbf{K}} =$

$\mathbf{K}_{\mathsf{M}}(\widehat{\boldsymbol{\theta}}) + \widehat{\mathbf{K}}_{\bar{\varepsilon}}$ in (19) and (20). Next we explain how we choose n_i.

We are interested in an experimental design with low "integrated MSPE" (IMSPE). Following Mehdad and Kleijnen (2015b) – who revised Ankenman et al. (2010) – we allocate (say) N replications to m old points $\mathbf{x}_i$ such that this design minimises the IMSPE. It is common practice to select these m points through a space-filling design; the most popular design uses Latin hypercube sampling (LHS). More specifically, let $\mathcal{X}$ be the design space. Then our design should result in

$$\min_{\mathbf{n}} \text{IMSPE}(\mathbf{n}) = \min_{\mathbf{n}} \int_{\mathbf{x}_0 \in \mathcal{X}} \text{MSPE}(\mathbf{x}_0, \mathbf{n}) d\mathbf{x}_0 \quad (22)$$

subject to $\mathbf{n}^\top \mathbf{1}_m \leq N$, and $\mathbf{n} = (n_1, \ldots, n_m)^\top$ where $n_i \in \mathbb{N}$. Mehdad and Kleijnen (2015b) derives the optimal allocation of the total number of replications N to the m old points

$$n_i^* = N \frac{\sqrt{\mathsf{V}(\mathbf{x}_i) C_i}}{\sum_{i=1}^m \sqrt{\mathsf{V}(\mathbf{x}_i) C_i}}, \text{ with}$$
$$C_i = \left[\mathbf{S}^{-1} \mathbf{W} \mathbf{S}^{-1}\right]_{p+1+i, p+1+i} \quad (23)$$

where

$$\mathbf{S} = \begin{bmatrix} \mathbf{O} & \mathbf{F}^\top \\ \mathbf{F} & \mathbf{K} \end{bmatrix},$$
$$\mathbf{W} = \int \begin{bmatrix} \mathbf{f}(\mathbf{x}_0)\mathbf{f}(\mathbf{x}_0)^\top & \mathbf{f}(\mathbf{x}_0)\mathbf{K}_{\mathsf{M}}(\mathbf{x}_0, \cdot)^\top \\ \mathbf{K}_{\mathsf{M}}(\mathbf{x}_0, \cdot)\mathbf{f}(\mathbf{x}_0)^\top & \mathbf{K}_{\mathsf{M}}(\mathbf{x}_0, \cdot)\mathbf{K}_{\mathsf{M}}(\mathbf{x}_0, \cdot)^\top \end{bmatrix} d\mathbf{x}_0.$$

Note that in (23) both the internal noise variance $\mathsf{V}(\mathbf{x})$ and the external noise covariance function K_{M} affect the allocation.

4. Efficient global optimisation (EGO)

In this section, we first summarise classic EGO and three variants; next we detail the variant that is the focus of our article. Classic EGO is developed by Jones et al. (1998) for deterministic simulation. It uses *expected improvement (EI)* as its criterion to balance local and global searches or exploiting and exploring. EGO is a *sequential* method with the following steps.

1. Fit a Kriging metamodel to the old I/O simulation data. Let $f_{\min} = \min_i Y(\mathbf{x}_i)$ be the minimum function value observed so far.
2. Estimate $\mathbf{x}_0$, which denotes the input combination that *maximises* $\text{EI}(\mathbf{x}) = \mathbb{E}\left[\max\left(f_{\min} - Y_p(\mathbf{x}), 0\right)\right]$. Assuming $Y_p(\mathbf{x}) \sim \mathcal{N}\left(\hat{Y}(\mathbf{x}), \hat{\sigma}^2(\mathbf{x})\right)$, Jones et al. (1998) derives

$$\widehat{\text{EI}}(\mathbf{x}) = \left(f_{\min} - \hat{Y}(\mathbf{x})\right) \Phi\left(\frac{f_{\min} - \hat{Y}(\mathbf{x})}{\hat{\sigma}(\mathbf{x})}\right) + \hat{\sigma}(\mathbf{x}) \phi\left(\frac{f_{\min} - \hat{Y}(\mathbf{x})}{\hat{\sigma}(\mathbf{x})}\right)$$

where $\hat{Y}(\mathbf{x})$ is defined in (4) and $\hat{\sigma}^2(\mathbf{x})$ follows from (5) substituting estimators for τ, and $\boldsymbol{\theta}$; Φ and ϕ denote the cumulative distribution function (CDF) and probability density function (PDF) of the standard normal distribution.
3. Simulate the response at $\hat{\mathbf{x}}_0$ found in step 2. Fit a new Kriging metamodel to the old and new I/O simulation data. Return to step 1, unless $\widehat{\text{EI}}$ satisfies a given criterion; e.g., $\widehat{\text{EI}}$ is less than 1% of the current best function value.

Huang et al. (2006) adapts EI for random simulation, using the metamodel defined in (16) and assuming that the noise variances are identical across the design space; i.e., $V(\mathbf{x}) = V$. EI is replaced by *augmented EI* (AEI):

$$\widehat{\text{AEI}}(\mathbf{x}) = \mathbb{E}\left[\max\left(\hat{Z}(\mathbf{x}^*) - Z_p(\mathbf{x}), 0\right)\right] \left[1 - \left(\frac{\hat{V}}{\text{MSPE}(\hat{Z}(\mathbf{x})) + \hat{V}}\right)^{1/2}\right] \quad (24)$$

where $\mathbf{x}^*$ is called the current 'effective best solution' and is defined as $\mathbf{x}^* = \arg\min_{\mathbf{x}_1, \ldots, \mathbf{x}_m} [\hat{Z}(\mathbf{x}) + \text{MSPE}(\hat{Z}(\mathbf{x}))]$; the second term on the right-hand side in (24) accounts for the diminishing returns of additional replications at the current best point.

Picheny et al. (2013) develops a quantile-based EI known as *expected quantile improvement* (EQI). This criterion lets the user specify the risk level; i.e., the higher the quantile is (e.g. the specified quantile is the 95% instead of the 90% quantile), the more conservative the criterion becomes. The EQI accounts for a limited computational budget; moreover, to sample a new point, EQI also considers the noise variance at future (not yet sampled) points. However, EQI requires a known variance function for the noise, and Quan et al. (2013) claims that Picheny et al. (2013)'s algorithm is also computationally more complex.

Quan et al. (2013) shows that EI and AEI cannot be good criteria for random simulations with heteroscedastic noise variances, because an EGO-type framework for random simulation with heteroscedastic noise faces three challenges: (1) An effective procedure should locate the global optimum with a limited computational budget. (2) To balance exploration and exploitation in random simulation, a new procedure should be able to search globally without exhaustively searching a local region; a good estimator of $f_{\min}$ is necessary especially, when there are several points that give outputs close to the global optimum. (3) With a limited computational budget, it is wise to obtain simulation observations in unexplored regions in the beginning of the search and – as the budget is being expended toward the end – focus on improving the current best area. Quan et al.'s algorithm addresses these challenge as follows: (1) the SK predictor is the underlying function that can

handle heteroscedastic noise. (2) The *Modified expected improvement* (MEI) criterion balances exploration and exploitation:

$$\widehat{\text{MEI}}(\mathbf{x}) = \mathbb{E}[\max(\hat{Z}_{\min} - Z_p(\mathbf{x}), 0)] \quad (25)$$

where $\hat{Z}_{\min}$ is the predicted response at the sampled point with the lowest sample mean, and Z_p is a normal random variable with the mean equal to $\hat{Z}(\mathbf{x})$ and the variance equal to the estimated MSPE($\hat{Z}(\mathbf{x})$). The MEI criterion uses only MSPE($\hat{Z}(\mathbf{x})$) with estimates of $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_M$ instead of $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_M + \boldsymbol{\Sigma}_{\bar{\varepsilon}}$. This helps the search to focus on the new points that reduce the spatial uncertainty of the metamodel. Ignoring the uncertainty caused by random variability, the MEI criterion assumes that the observations are made with infinite precision so the same point is never selected again. This helps the algorithm to quickly escape from a local optimum, and makes the sampling behaviour resemble the behaviour of the original EI criterion with its balancing of exploration and exploitation. To have a good estimator of $f_{\min}$, OCBA distributes an additional number of replications over the old sampled points and the new sampled point that maximises MEI. (3) The computational budget per iteration is set as a constant, but the allocation of this budget between the searching stage and the allocation stage changes as the algorithm progresses. In the beginning, most of the budget is invested in exploration (search stage). During the progress of the algorithm, the focus moves to identifying the point with the lowest sample mean (allocation stage). We give the details of Quan et al.'s algorithm in Appendix 1.

We introduce a new algorithm that differs from Quan et al.'s algorithm in two ways: (i) For the underlying metamodel, we use SIK instead of SK. (ii) In the allocation stage, we use (23) instead of OCBA to distribute an additional number of replications to the old and new sampled points. We do use Quan et al.'s MEI defined in 25 in our algorithm. We call our new algorithm *stochastic efficient global optimisation* (SEGO).

5. Numerical experiments

We experiment with deterministic simulation and random simulation. In these experiments we use a zero-degree polynomial for the trend (so $p = 0$ in (1)), so UK becomes *ordinary Kriging* (OK). Mehdad and Kleijnen (2015b) suggests that the best GCF candidate for IK is the *integrated Brownian motion* covariance function defined in (8). In OK and SK, we select the covariance functions that are most popular in simulation; namely, Gaussian covariance functions. For deterministic simulation, we study the performance of EGO with OK versus EGO with IK. For random simulation, we study the performance of Quan et al.'s algorithm versus our SEGO algorithm with SIK (instead of SK) and the minimum IMSE allocation rule (instead of OCBA).

We tried to use the MATLAB code developed by Yin et al. (2011) – which is a building block in Quan et al.'s algorithm – to experiment with the Kriging variants (OK for deterministic simulation and SK for random simulation), but that MATLAB code crashed in experiments with $d > 1$. We therefore use the R package `DiceKriging` to implement OK and SK; for details on `DiceKriging` we refer to Roustant, Ginsbourger, and Deville (2012). We implement our code for IK and SIK in MATLAB, as Mehdad and Kleijnen (2015b) and Mehdad (2015) also do. In all our experiments we select $\mathbf{k} = \mathbf{0}$, because this choice gives the lowest MSE metamodel in most experiments in Mehdad and Kleijnen (2015b). Note that setting $\mathbf{k} = 0$ in (8) makes $K(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$ become $\prod_{g=1}^{d} \left(\theta_{0;g} + \theta_{1;g} \min\left\{x_g, x'_g\right\}\right)$.

Furthermore, we select a set of $m_c = 100d$ candidate points to estimate the point that maximises EI or MEI. For $d = 1$ we select m_0 and m_c equispaced points; for $d > 1$ we select m_0 and m_c space-filling points through LHS, implemented in the MATLAB function `lhsdesign`.

As the criterion for comparing the performance of different optimisation algorithms, we use the number of simulated input combinations needed to estimate the optimal input combination (say) m. As the stopping criterion, we select m reaching a limit; namely, 11 for $d = 1$, 61 for the camel-back test function ($d = 2$), 65 for Hartmann-3 ($d = 3$), and 111 for Ackley-5 ($d = 5$); we define the first test function in (26) and the other three test functions in the appendix. We select the number of starting points m_0 to be 3 for $d = 1$, 21 for $d = 2$, 30 for $d = 3$, and 51 for $d = 5$.

5.1. Deterministic simulation experiments

We experiment with several multi-modal test functions of different dimensionality. We start with Gramacy and Lee (2012)'s test function with $d = 1$:

$$f(x) = \frac{\sin(10\pi x)}{2x} + (x-1)^4 \text{ and } 0.5 < x \leq 2.5. \quad (26)$$

Next we experiment with the other three functions with $d > 1$ that are often used as test functions in optimisation; see Dixon and Szego (1978) and http://www.sfu.ca/~ssurjano/index.html.

Figure 1 illustrates seven iterations of EGO with IK for the $d = 1$ test function defined in (26). This function has a global minimum at $x_{\text{opt}} = 0.5486$ with output $f(x_{\text{opt}}) = -0.869$; also see the curves in the left panels of the figure, where the (blue) solid curve is the true function and the (red) dotted line is the IK metamodel. We start with $m = 3$ old points, and stop after sequentially adding seven new points (shown by black circles); i.e., from top to bottom the number of

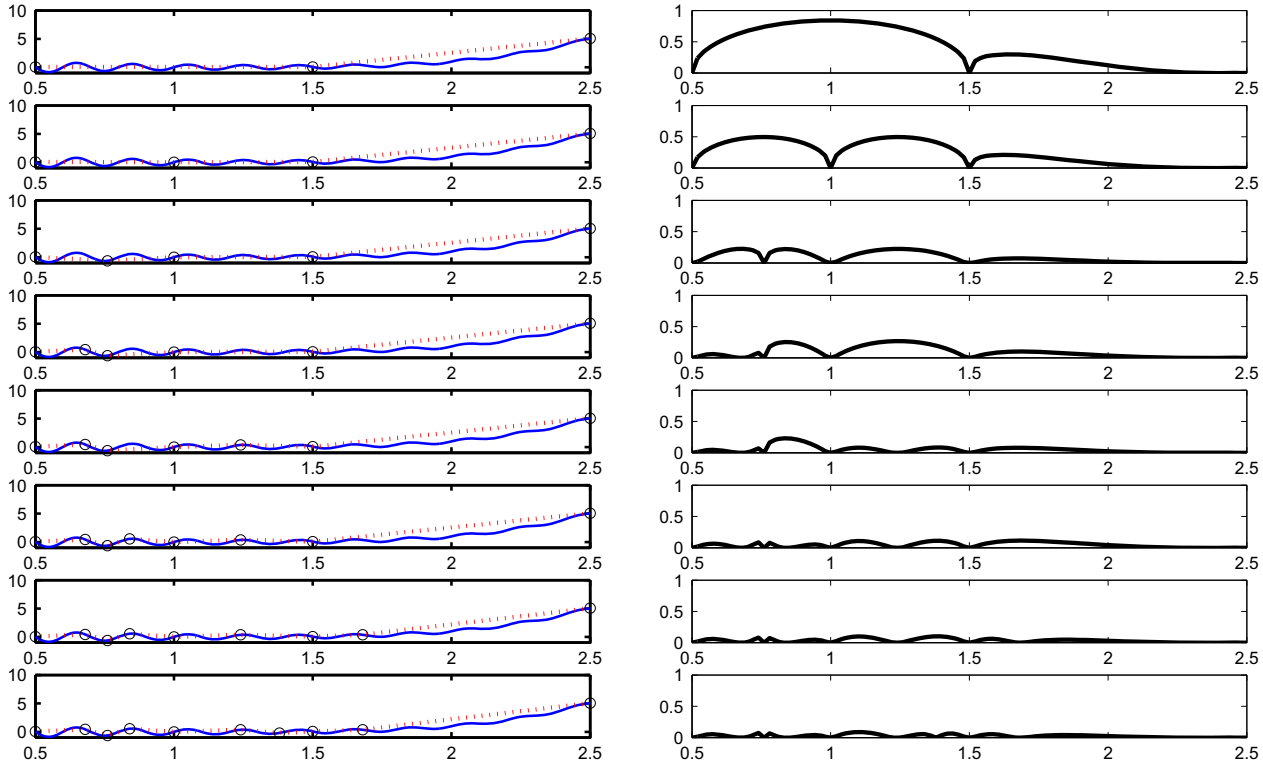

Figure 1. EGO with IK for Gramacy and Lee (2012)'s test function.

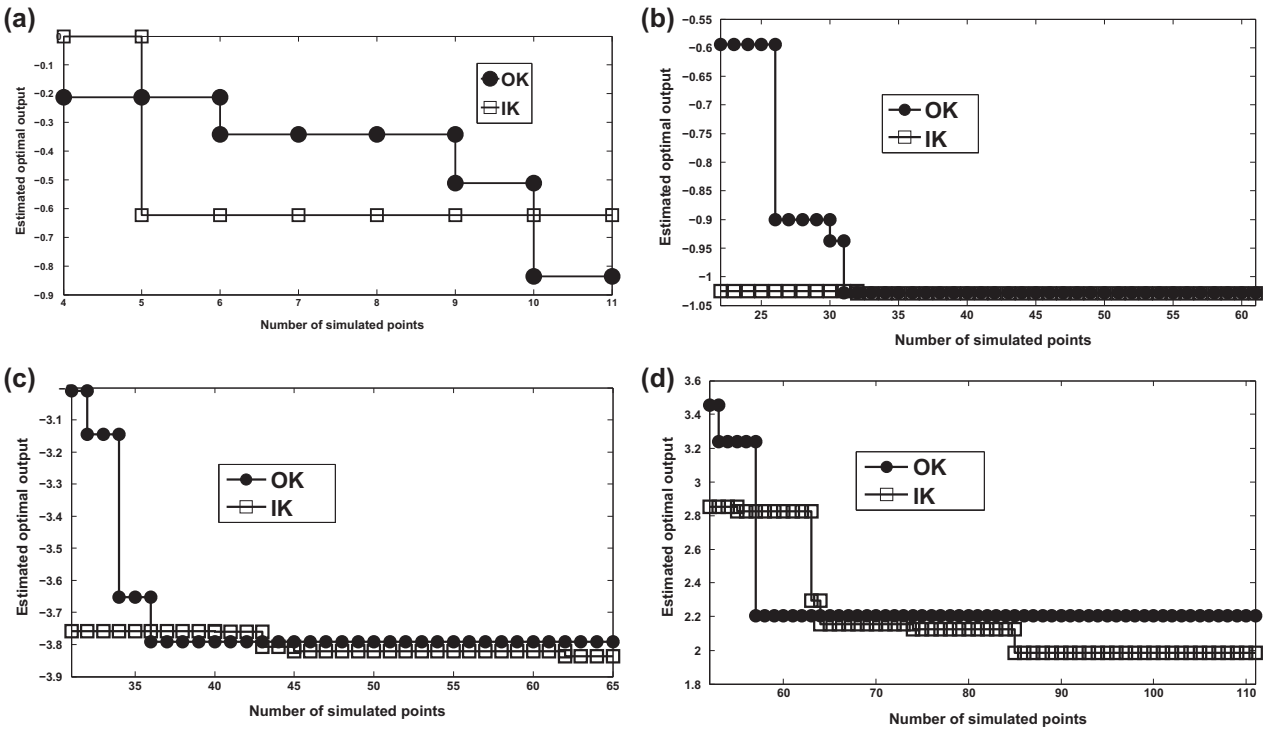

Figure 2. Estimated optimal output (y-axis) after m simulated input combinations (x-axis) in four test functions, for deterministic simulation. (a) $\widehat{\text{EI}}$ with IK and OK in Gramacy; (b) $\widehat{\text{EI}}$ with IK and OK in camel-back; (c) $\widehat{\text{EI}}$ with IK and OK in Hartmann-3; (d) $\widehat{\text{EI}}$ with IK and OK in Ackley-5.

points increases starting with three points and ending with ten points. These plots show that a small m gives a poor metamodel. The right panels display $\widehat{\text{EI}}$ as m increases. These panels show that $\widehat{\text{EI}} = 0$ at the old points.

Figure 2 displays $f_{\min}(m) = \min f(\hat{\mathbf{x}}_i)$ $(1 \leq i \leq m)$, which denotes the estimated optimal simulation output

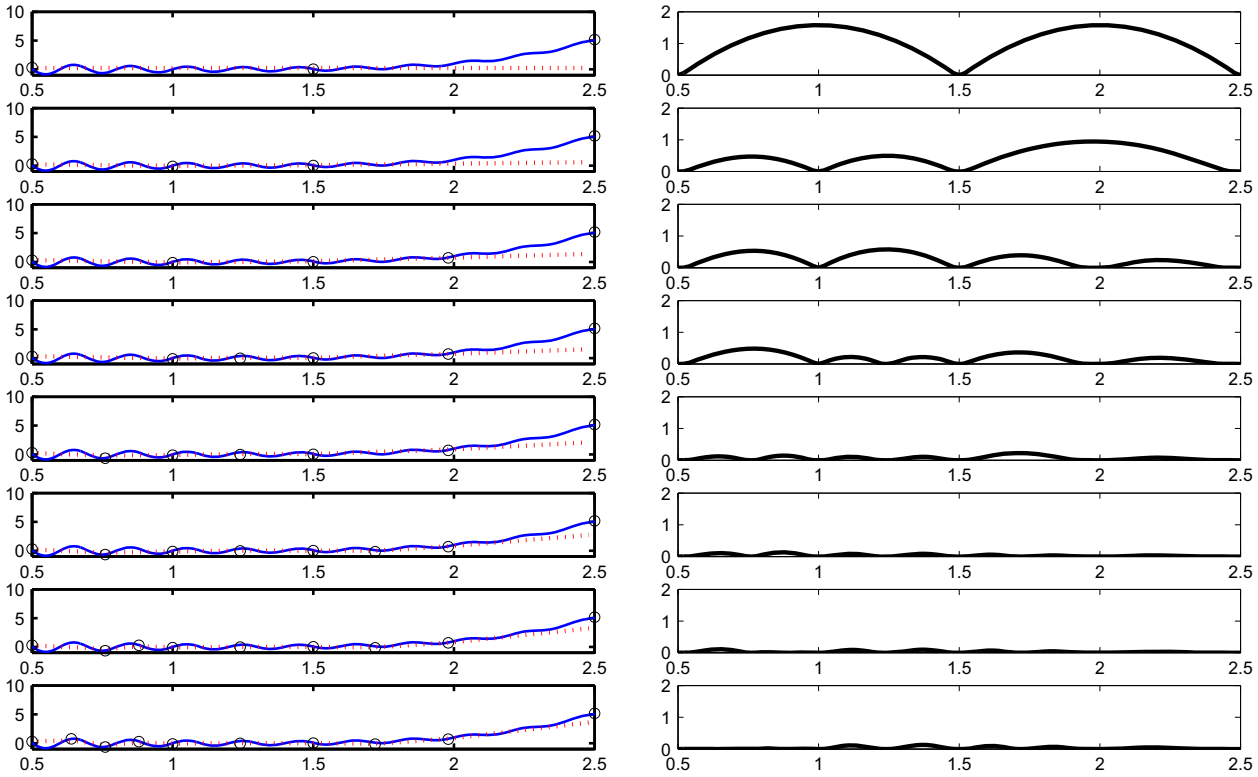

Figure 3. SEGO algorithm for Gramacy and Lee (2012)'s function.

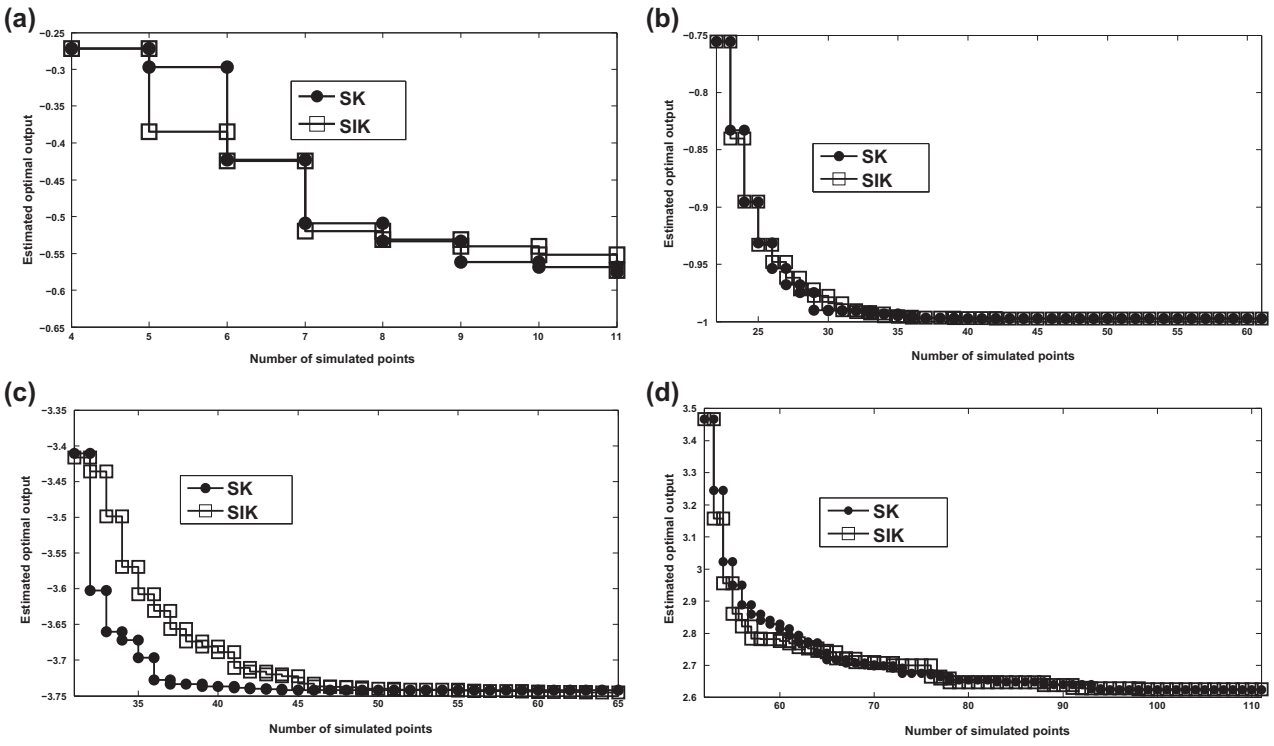

Figure 4. Estimated optimal output (y-axis) after m simulated input combinations (x-axis) for four test functions, for random simulation. (a) $\widehat{\text{MEI}}$ with SIK and SK in Gramacy; (b) $\widehat{\text{MEI}}$ with SIK and SK in camel-back; (c) $\widehat{\text{MEI}}$ with SIK and SK in Hartmann-3; (d) $\widehat{\text{MEI}}$ with SIK and SK in Ackley-5.

after m simulated input combinations; horizontal lines mean that the most recent simulated point does not give a lower estimated optimal output than a preceding point. The square marker represents EGO with IK, and the circle marker represents classic EGO with OK. The results show that in most experiments, EGO with IK performs better than classic EGO with OK; i.e., EGO with IK gives a better input combination after fewer simulated points. For example, in part (a) of this Figure, IK finds much better solutions than OK does – until the

sample size becomes relatively big; namely, bigger than 10.

5.2. Random simulation experiments

Now, we compare the performance of Quan et al.'s algorithm with our SEGO algorithm that uses SIK as the metamodel and a different allocation rule. In both algorithms, we select the minimum number of replications for a new point $r_{\min} = 10$ and the number of replications available for each iteration $B = 40$ (for $d = 1$), 130 (for $d = 2$ and 3), and 310 (for $d = 5$). In all our experiments for random simulation, we augment the deterministic test functions defined for our deterministic experiments, with the heteroscedastic noise $\mathsf{V}(\mathbf{x}_i) = (1 + |y(\mathbf{x}_i)|)^2$.

Figure 3 illustrates seven iterations of our SEGO algorithm for the $d = 1$ test function. We start with $m = 3$ old points, and stop after sequentially adding seven new points. With small m and high noise ($\mathsf{V}(x)$ increases as x increases), the metamodel turns out to be a "poor" approximation: compare the (blue) solid curves and the (red) dashed curves. Note that in the beginning the algorithm searches the region far from the unknown global optimum (namely, $x = 0.5486$), and after each iteration and careful allocation of added replications, the quality of the SIK fit in areas with high noise (as x increases) improves. The right panels display $\widehat{\text{MEI}}$ as m increases.

Finally, we compare the two algorithms using 50 macro-replications. Figure 4 displays $f_{\min}(m) = \min_i \left(\sum_{t=1}^{50} f_t(\hat{\mathbf{x}}_i) \Big/ 50\right)$, $1 \le i \le m$, which denotes the estimated optimal simulation output averaged over 50 macro-replications. The circle marker represents Quan et al.'s original algorithm, and the square marker represents our SEGO algorithm. The results show that the two algorithms do not give significantly different results in most sampled points, except for Hartmann-3 where Quan et al.'s algorithm performs significantly better for $m = 32, \ldots, 45$ and our SEGO performs significantly better for $m = 31, 59, \ldots, 65$. We note that this conclusion is confirmed by paired t-tests, which we do not detail here.

6. Conclusions

We modified Jones et al. (1998)'s classic EGO replacing OK by IK for deterministic simulation; for stochastic simulation, we modified Quan et al.'s algorithm replacing SIK by SK and introducing a new allocation rule. We quantified the efficiency and effectiveness of the various algorithms through numerical experiments with several classic test functions. Our main conclusion is that in most experiments; (i) in deterministic simulations, EGO with IK performs better than classic EGO with OK; (ii) in random simulation, there is no significant performance difference between Quan et al.'s algorithm and our SEGO algorithm.

In future research we may further investigate the choice of the algorithm's parameter values, and the robustness of the algorithm's performance with respect to the choices of these parameters. Furthermore, we may investigate the allocation of replications in random simulation analysed through Kriging metamodels. Finally, we may investigate more test functions (besides the four functions that we used), and practical applications of our methodology.

Acknowledgements

We thank two anonymous reviewers for their very useful comments on a previous version.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Ankenman, B., Nelson, B., & Staum, J. (2010). Stochastic Kriging for simulation metamodeling. *Operations Research, 58*, 371–382.

Chen, C., Lin, J., Yucesan, E., & Chick, S. (2000). Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems, 10*(3), 251–270.

Chilès, J., & Delfiner, P. (2012). *Geostatistics: Modeling spatial uncertainty*. New York, NY: Wiley.

Cressie, N. (1991). *Statistics for spatial data*. New York, NY: Wiley.

(1978). Dixon, L., & Szego, G.. eds. *Towards global optimisation 2*. North Holland, Amsterdam: Elsevier Science Ltd.

Ginsbourger, D. (2009). Multiples Métamodèles pour l'Approximation et l'Optimisation de Fonctions Numériques Multivariables (PhD dissertation), Ecole Nationale Supérieure des Mines de Saint-Etienne.

Gramacy, R., & Lee, H. (2012). Cases for the nugget in modeling computer experiments. *Statistics and Computing, 22*, 713–722.

Hoad, K., Monks, T., & O'Brien, F. (2015). The use of search experimentation in discrete-event simulation practice. *Journal of the Operational Research Society, 66*, 1155–1168.

Hoad, K., Robinson, S., & Davies, R. (2010). Automated selection of the number of replications for a discrete-event simulation. *Journal of the Operational Research Society, 61*(11), 1632–1644.

Huang, D., Allen, T., Notz, W., & Zeng, N. (2006). Global optimization of stochastic black-box systems via sequential Kriging meta-models. *Journal of Global Optimization, 34*, 441–466.

Jones, D., Schonlau, M., & Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization, 13*, 455–492.

Kleijnen, J. (2015). *Design and analysis of simulation experiments*. 2nd ed.New York, NY: Springer-Verlag.

Loeppky, J., Sacks, J., & Welch, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics, 51*, 366–376.

Mehdad, E. (2015). *Kriging metamodels and global opimization in simulation.* (CentER PhD Dissertation Series)Tilburg University, School of Economics and Management.

Mehdad, E. & Kleijnen, J. (2015a). Classic Kriging versus Kriging with bootstrapping or conditional simulation: Classic Krigings robust confidence intervals and optimization. *Journal of the Operational Research Society, 66*, 1804–1814.

Mehdad, E., & Kleijnen, J. (2015b). *Stochastic intrinsic Kriging for simulation metamodelling* (CentER Discussion Paper No. 2015-038). Tilburg University.

Opsomer, J., Ruppert, D., Wand, M., Holst, U., & Hossjer, O. (1999). Kriging with nonparametric variance function estimation. *Biometrics, 55*(3), 704–710.

Picheny, V., Ginsbourger, D., Richet, Y., & Caplin, G. (2013). Quantile-based optimization of noisy computer experiments with tunable precision (including comments). *Technometrics, 55*(1), 2–36.

Quan, N., Yin, J., Ng, S., & Lee, L. (2013). Simulation optimization via Kriging: A sequential search using expected improvement with computing budget constraints. *IIE Transactions, 45*(7), 763–780.

Roustant, O., Ginsbourger, D., & Deville, Y. (2012). DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization. *Journal of Statistical Software, 51*(1), 1–55.

Salemi, P., Nelson, B. & Staum, J. (2014). Discrete optimization via simulation using Gaussian Markov random fields. In *Proceedings of the 2014 Winter Simulation Conference (WSC)* (pp. 3809–3820). IEEE Press.

Sun, L., Hong, L., & Hu, Z. (2014). Balancing exploitation and exploration in discrete optimization via simulation through a Gaussian process-based search. *Operations Research, 62*(6), 1416–1438.

Yin, J., Ng, S., & Ng, K. (2011). Kriging meta-model with modified nugget effect: an extension to heteroscedastic variance case. *Computers and Industrial Engineering, 61*(3), 760–777.

Appendix 1. Quan's algorithm

After the initial fit of a SK metamodel, each iteration of the algorithm consists of a search stage followed by an allocation stage. In the search stage, the MEI criterion is used to select a new point; see (25). Next in the allocation stage, OCBA distributes an additional number of replications among the sampled points. These additional replications are distributed with the goal of maximising the *probability of correct selection* (PCS) when selecting a sampled point as the global optimum. Suppose we have m sampled points $\mathbf{x}_i$ $(i = 1, \ldots, m)$ with sample mean $\overline{Z}_i$ and sample variance $\widehat{V}(\mathbf{x}_i) = s^2(\mathbf{x}_i)$. Then the *approximate PCS* (APCS) can be asymptotically maximised when the available computational budget N tends to infinity and

$$\frac{n_i}{n_j} = \left(\frac{\widehat{V}(\mathbf{x}_i)/\Delta_{b,i}}{\widehat{V}(\mathbf{x}_j)/\Delta_{b,j}}\right)^2 \quad i,j = 1,\ldots,m \text{ and } i \neq j \neq b, \tag{A1}$$

$$n_b = \widehat{V}(\mathbf{x}_b)\sqrt{\sum_{i=1,i\neq b}^{m} \frac{n_i^2}{\widehat{V}(\mathbf{x}_i)}} \tag{A2}$$

where n_i is the number of replications allocated to $\mathbf{x}_i$, $\mathbf{x}_b$ is the point with the lowest sample mean, and $\Delta_{b,i}$ is the difference between the lowest sample mean and the sample mean at point $\mathbf{x}_i$; Chen, Lin, Yucesan, and Chick (2000) proves (A1) and (A2). Given this allocation rule, the sampled point with the lowest sample mean will be selected as $\hat{Z}_{\min}$ at the end of the allocation stage.

Algorithm 1 Quan et al.'s algorithm

Step 1. Initialization: Run a space filling design with m_0 points, with B replications allocated to each point.
Step 2. Validation: Fit a SK metamodel to the set of sample means. Use leave-one-out cross validation to check the quality of the initial SK.
Step 3. Set $i = 1, r_A(0) = 0$
while $i \leq I$ **do**

$$r_A(i) = r_A(i-1) + \min\left(\lfloor\frac{B - r_{\min}}{I}\rfloor, T - m_0B - (i-1)B\right)$$

if $(T - m_0B - (i-1)B - r_A(i)) > 0$ **then**

$r_S(i) = B - r_A(i)$
Step 3a. Search Stage: Sample a new point that maximises the MEI criterion (Equation 25) with $r_S(i)$ replications.
Step 3b. Allocation Stage: Using OCBA (Equations A1 and A2), allocate $r_A(i)$ replications among all sampled points.
Step 3c. Fit a SK metamodel to the set of sample means
$i = i + 1$

end if

end while
The point with the lowest sample mean at the end estimates the global optimum.

We summarise the algorithm in our Algorithm 1. Before the algorithm begins, the user must specify T, B, m_0, and $r_{\min}$ where T is the total number of replications at the start, B is the number of replications available for each iteration, m_0 is the size of the initial space filling design, and $r_{\min}$ is the minimum number of replications for a new point. The size of the initial design m_0 may be set to $10d$ where d is the number of dimensions ($10d$ is proposed by Loeppky, Sacks, and Welch (2009)); B and $r_{\min}$ should be set such that there are sufficient replications available for the first allocation stage.

The starting parameters that determine the number of iterations, $I = \lceil (T - m_0B)/B \rceil$, and the computational budget used per iteration B are set prior to collecting any data, so the starting parameter settings may turn out to be unsuitable for the problem. In step 2, leave-one-out cross validation can provide feedback regarding the suitability of the initial parameters. If one or more design points fail the cross-validation test, then the computational budget may be insufficient to deal with the internal noise. Possible solutions include (i) increasing B, (ii) increasing the number of design points around the point(s) that fail the cross-validation test, and (iii) applying a logarithmic or inverse transformation to the simulation response.

After successful validation of the SK metamodels, the computational budget set aside for the allocation stage $r_A(i)$ increases by a block of $\lfloor (B - r_{\min})/I \rfloor$ replications in every iteration while $r_S(i)$ decreases by the same amount for the search stage. This heuristic gives the algorithm the desirable characteristic of focusing on exploration at the start and on exploitation at the end of the search.

Appendix 2. Three test functions with dimensionality $d > 1$

We define three test functions with $d > 1$.

1. Camel-back function with $-2 \leq x_1 \leq 2$, $-1 \leq x_2 \leq 1$, $\mathbf{x}_{\text{opt}}^\top = (\pm 0.0898, \mp 0.7126)$, and $f(\mathbf{x}_{\text{opt}}) = -1.0316$

$$f(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4.$$

2. Hartmann-3 function with $0 \leq x_i \leq 1$, $i = 1, 2, 3$, $\mathbf{x}_{\text{opt}}^\top = (0.114614, 0.555649, 0.852547)$, and $f(\mathbf{x}_{\text{opt}}) = -3.86278$

$$f(x_1, x_2, x_3) = -\sum_{i=1}^{4} \alpha_i \exp[-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2]$$

Table B1. Parameters A_{ij} and P_{ij} of the Hartmann-3 function.

	A_{ij}			P_{ij}	
3	10	30	0.36890	0.1170	0.26730
0.1	10	35	0.46990	0.43870	0.74700
3	10	30	0.10910	0.87320	0.55470
0.1	10	35	0.03815	0.57430	0.88280

with $\boldsymbol{\alpha} = (1.0, 1.2, 3.0, 3.2)^\top$ and A_{ij} and P_{ij} given in Table B1.

3. Ackley-5 function with $-2 \leq x_i \leq 2,\ i = 1, \ldots, 5$, and $(\mathbf{x}_{\text{opt}} = \mathbf{0}, f(\mathbf{x}_{\text{opt}}) = 0)$

$$f(\mathbf{x}) = -20 \exp\left(-0.2\sqrt{\frac{1}{5}\sum_{i=1}^{5} x_i^2}\right) - \exp\left(\frac{1}{5}\sum_{i=1}^{5}\cos(2\pi x_i)\right) + 20 + \exp(1).$$