Computational Statistics and Data Analysis 94 (2016) 250–264

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models

Joshua Svenson [a,*], Thomas Santner [b]

[a] JPMorgan Chase & Co., 1111 Polaris Parkway, Columbus, OH, 43240, USA

[b] Department of Statistics, The Ohio State University, 1958 Neil Avenue, Columbus, OH, 43210, USA

ARTICLE INFO

Article history:
Received 20 June 2014
Received in revised form 6 May 2015
Accepted 22 August 2015
Available online 3 September 2015

Keywords:
Computer experiment
Gaussian process
Kriging
Pareto optimization
Nonseparable GP model
Computer simulator model

ABSTRACT

Many engineering design optimization problems contain multiple objective functions all of which are desired to be minimized, say. This paper proposes a method for identifying the Pareto Front and the Pareto Set of the objective functions when these functions are evaluated by *expensive-to-evaluate* deterministic computer simulators. The method replaces the expensive function evaluations by a rapidly computable approximator based on a Gaussian process (GP) interpolator. It sequentially selects new input sites guided by values of an "improvement function" given the current data. The method introduced in this paper provides two advances in the interpolator/improvement framework. First, it proposes an improvement function based on the "modified maximin fitness function" which is known to identify well-spaced non-dominated outputs when used in multiobjective evolutionary algorithms. Second, it uses a family of GP models that allows for dependence among output function values but which permits zero covariance should the data be consistent with this model. A closed-form expression is derived for the improvement function when there are two objective functions; simulation is used to evaluate it when there are three or more objectives. Examples from the multiobjective optimization literature are presented to show that the proposed procedure can improve substantially previously proposed statistical improvement criteria for the computationally intensive multiobjective optimization setting.

1. Introduction

This paper proposes an algorithm for sequentially designing a sequence of inputs at which to run a set of expensive-to-evaluate functions so as to identify the Pareto Front of function values and the Pareto Set of inputs. Throughout, let $\boldsymbol{y}(\boldsymbol{x}) = (y_1(\boldsymbol{x}), \ldots, y_m(\boldsymbol{x}))$ denote the m functions of interest, d the number of inputs, and $\boldsymbol{x} = (x_1, \ldots, x_d)$ a generic input. The input space for $\boldsymbol{x}$ is denoted by $\mathcal{X} \subset \mathbb{R}^d$ and the function values $y(\boldsymbol{x})$, $\boldsymbol{x} \in \mathcal{X}$, form the *objective space*.

The goal of this paper is to find the *complement* of the set inputs $\boldsymbol{x} \in \mathcal{X}$ that are dominated by one or more inputs in $\mathcal{X}$. An input $\boldsymbol{x}_1 \in \mathcal{X}$ *weakly dominates* $\boldsymbol{x}_2 \in \mathcal{X}$ ($\boldsymbol{x}_1 \succeq \boldsymbol{x}_2$) if $y_i(\boldsymbol{x}_1) \leq y_i(\boldsymbol{x}_2)$ for all $i = 1, \ldots, m$. If at least one inequality is strict, then $\boldsymbol{x}_1$ is said to *dominate* $\boldsymbol{x}_2$ ($\boldsymbol{x}_1 \succ \boldsymbol{x}_2$). Equivalently, an input $\boldsymbol{x}_1$ does *not dominate* $\boldsymbol{x}_2$ ($\boldsymbol{x}_1 \not\succ \boldsymbol{x}_2$) if there exists any i such that $y_i(\boldsymbol{x}_1) > y_i(\boldsymbol{x}_2)$. Geometrically, $\boldsymbol{x}_1 \succ \boldsymbol{x}_2$ if $\boldsymbol{y}(\boldsymbol{x}_1)$ is strictly to the "southwest" of $\boldsymbol{y}(\boldsymbol{x}_2)$.

In an analogous fashion, for $\boldsymbol{y}(\boldsymbol{x}_1)$ and $\boldsymbol{y}(\boldsymbol{x}_2)$ in the objective space, $\boldsymbol{y}(\boldsymbol{x}_1)$ is said to *weakly dominate* $\boldsymbol{y}(\boldsymbol{x}_2)$ ($\boldsymbol{y}(\boldsymbol{x}_1) \succeq \boldsymbol{y}(\boldsymbol{x}_2)$) if $y_i(\boldsymbol{x}_1) \leq y_i(\boldsymbol{x}_2)$ for all $i = 1, \ldots, m$. If at least one inequality is strict, then $\boldsymbol{y}(\boldsymbol{x}_1)$ is said to *dominate* $\boldsymbol{y}(\boldsymbol{x}_2)$ ($\boldsymbol{y}(\boldsymbol{x}_1) \succ \boldsymbol{y}(\boldsymbol{x}_2)$).

* Corresponding author.
E-mail addresses: joshua.d.svenson@chase.com (J. Svenson), santner.1@osu.edu (T. Santner).

http://dx.doi.org/10.1016/j.csda.2015.08.011
0167-9473/

An output $\boldsymbol{y}(\boldsymbol{x}_1)$ does *not dominate* output $\boldsymbol{y}(\boldsymbol{x}_2)$ ($\boldsymbol{y}(\boldsymbol{x}_1) \not\succ \boldsymbol{y}(\boldsymbol{x}_2)$) if there exists any i such that $y_i(\boldsymbol{x}_1) > y_i(\boldsymbol{x}_2)$. An input vector $\boldsymbol{x} \in \mathcal{X}$ is *Pareto optimal* if and only if there is no $\boldsymbol{x}' \in \mathcal{X}$ such that $\boldsymbol{x} \prec \boldsymbol{x}'$. (Such $\boldsymbol{x}$ are also referred to as *nondominated inputs*. The image $\boldsymbol{y}(\boldsymbol{x})$ of a nondominated input is sometimes referred to as a *nondominated output*.)

Thus the goal can be restated as that of finding the set of all $\boldsymbol{x} \in \mathcal{X}$ which are not dominated by any other input in $\mathcal{X}$; the set of nondominated inputs is called the *Pareto Set*. The set of $(y_1(\boldsymbol{x}), \ldots, y_m(\boldsymbol{x}))$ corresponding to $\boldsymbol{x}$ in the Pareto Set is termed the *Pareto Front*. This paper proposes an algorithm that uses previous $y(\boldsymbol{x})$ evaluations to determine a sequence of inputs $\boldsymbol{x}$ to identify the Pareto Set and the associated Pareto Front.

In most real-world applications, the Pareto Front is an uncountable set and cannot be found analytically. Therefore this paper, as do virtually all papers that identify Pareto Fronts/Sets, finds a discrete approximation to the Pareto Front. In addition, many current methodologies for approximating Pareto Fronts and Pareto Sets, such as the weighted sum method, the ϵ-constrained method, multiobjective evolutionary algorithms (see Coello et al., 2006), and the MULTIMADS algorithm introduced in Audet et al. (2010) have been designed for applications where many (hundreds, possibly thousands) of function evaluations are feasible. Under these conditions, the algorithms above have proven to be very effective at identifying these two Pareto Sets. However, in multiobjective settings where one is running a detailed deterministic computer simulator that is expensive-to-evaluate, only a few dozen function evaluations may be available. This paper proposes methodology for such cases.

In overview, the proposed methodology builds a rapidly-computable surrogate for $\boldsymbol{y}(\boldsymbol{x})$, which is used to guide the search for nondominated points. The $\boldsymbol{y}(\boldsymbol{x})$ surrogate that the authors employ is an interpolator of the training data based on a *Gaussian process* (GP) stochastic model (see Santner et al., 2003). The search selects the $\boldsymbol{x} \in \mathcal{X}$ which maximizes a heuristically selected "expected improvement" criterion. The methodology proposed in this paper provides two key improvements over other interpolator/expected improvement schemes that have been considered in the literature (Schonlau, 1997; Jones et al., 1998; Keane, 2006; Emmerich et al., 2006; Knowles, 2006), First, it is the only proposed multiobjective expected improvement approach that considers stochastic prediction models which allow for dependence *among* the components of $\boldsymbol{y}(\boldsymbol{x})$; procedures that have used dependence models in other applications can lead to improved procedure performance (Ver Hoef and Cressie, 1993; Williams et al., 2010; Fricker et al., 2013). Second, the improvement criterion is based on the maximin fitness function (see Balling, 2003) from the multiobjective evolutionary algorithm (MOEA) literature. This metric of distance quantifies how much better a given output vector is than the current best estimate of the Pareto front and directs MOEAs towards well-spaced designs that are close to the true Pareto front.

The remainder of this paper is organized as follows. To provide context, Section 2 reviews the expected improvement approach proposed in Schonlau (1997) and Jones et al. (1998) for single-objective functions. Section 3 describes the multivariate Gaussian process model that forms the basis for the proposed objective function emulators. Section 4 introduces the proposed improvement criterion and describes its implementation. Section 5 presents the sequential algorithm used to approximate Pareto Front and Pareto Set. Section 6 presents two examples that contrasts the new method with previous proposals from Keane (2006). Finally, Section 7 contains recommendations as to which methods should be used in practice, compares the proposed approach to the hypervolume-based method of Emmerich et al. (2006), and discusses future research regarding the expected improvement approach to multiobjective optimization.

2. Optimization of a single black-box function

To facilitate understanding the multivariate optimization proposal given in this paper, this section introduces the key ideas for the simpler problem of minimizing a single (expensive-to-evaluate) real-valued function $y(\boldsymbol{x})$ defined on a d-dimensional input space $\mathcal{X}$. The method described is due to Schonlau (1997), Jones et al. (1998) who introduced a methods for minimizing $y(\cdot)$ based on a GP model which they called the "efficient global optimization" (EGO) algorithm. The EGO algorithm uses a probabilistic assessment of $y(\boldsymbol{x})$ given the current data that is provided by the GP model. Specifically, these authors determine the information about the global minimum of $y(\cdot)$ that is in each potential $\boldsymbol{x}$ by the (conditional) expectation of a heuristically selected improvement function.

Suppose that $y(\cdot)$ has been evaluated at each input in the "design" $\mathcal{D}_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathcal{X}$. Let $\boldsymbol{y}^n = (y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_n))^T$ denote the corresponding vector of outputs. The deterministic output $y(\boldsymbol{x})$ is regarded as a draw from a stationary GP, $Y(\boldsymbol{x})$, with mean β, variance σ^2, and correlation function

$$Cov\left(Y(\boldsymbol{x}), Y(\boldsymbol{x}^\star)\right) = R\left(\boldsymbol{x} - \boldsymbol{x}^\star \mid \boldsymbol{\theta}\right) = \exp\left\{-\sum_{i=1}^{d} \theta_i \left(x_i - x_i^\star\right)^2\right\}, \tag{1}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$. The parameters $(\beta, \sigma^2, \boldsymbol{\theta})$ are unknown and must be estimated to complete specification of the GP model.

The GP provides the basis for interpolation of $y(\cdot)$ and uncertainty assessment of the predicted values. When $\left(\sigma^2, \boldsymbol{\theta}\right)$ is known, the best linear unbiased predictor (BLUP) of $y(\boldsymbol{x}_0)$ is

$$\widehat{y}(\boldsymbol{x}_0) = \widehat{\beta} + \boldsymbol{r}^\top(\boldsymbol{x}_0)\boldsymbol{R}^{-1}\left(\boldsymbol{y}^n - \mathbf{1}\widehat{\beta}\right), \tag{2}$$

Sacks et al. (1989). Here $\boldsymbol{R} = (R_{ij})$ is the $n \times n$ matrix with $R_{ij} = R\left(\boldsymbol{x}_i - \boldsymbol{x}_j \mid \boldsymbol{\theta}\right)$, $\boldsymbol{r}(\boldsymbol{x}_0) = (r_i(\boldsymbol{x}_0))$ is the $n \times 1$ vector with $r_i(\boldsymbol{x}_0) = R\left(\boldsymbol{x}_0 - \boldsymbol{x}_i \mid \boldsymbol{\theta}\right)$, and

$$\widehat{\beta} = \left(\mathbf{1}^\top \boldsymbol{R}^{-1} \boldsymbol{y}^n\right) / \left(\mathbf{1}^\top \boldsymbol{R}^{-1} \mathbf{1}\right). \tag{3}$$

Alternatively, $\widehat{y}(\boldsymbol{x}_0)$ is the mean of $Y(\boldsymbol{x}_0)$ conditional on $Y^n = (Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n)) = \boldsymbol{y}^n$ when $\boldsymbol{\beta}$ has a diffuse prior and other parameters are given. Indeed, conditional on $\boldsymbol{y}^n$, $Y(\boldsymbol{x}_0)$ is normally distributed with mean $\widehat{y}(\boldsymbol{x}_0)$ and variance

$$s^2(\boldsymbol{x}_0) \equiv E\{(Y(\boldsymbol{x}_0) - \widehat{y}(\boldsymbol{x}_0))^2 | \boldsymbol{y}^n\} = \sigma^2 \left[1 - \boldsymbol{r}^\top(\boldsymbol{x}_0)\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}_0) + \frac{\left(1 - \mathbf{1}^\top \boldsymbol{R}^{-1}\mathbf{1}\right)^2}{\mathbf{1}^\top \boldsymbol{R}^{-1}\mathbf{1}}\right]. \tag{4}$$

The quantity $s^2(\boldsymbol{x}_0)$ is the mean square prediction error (MSPE) of $\widehat{y}(\boldsymbol{x}_0)$ is thus used to quantify the uncertainty of this predictor.

In practice, $\left(\sigma^2, \boldsymbol{\theta}\right)$ is unknown. The frequentist approach to modifying (2) is to estimate $\left(\sigma^2, \boldsymbol{\theta}\right)$ by, say, maximum likelihood and apply (2) and (4) yielding an "empirical" BLUP, $\widehat{y}(\boldsymbol{x}_0)$, and estimated MSPE, $s^2(\boldsymbol{x}_0)$. An alternative approach to modifying (2) when $\left(\sigma^2, \boldsymbol{\theta}\right)$ is unknown is the Bayesian paradigm in which priors are identified that embody knowledge about $\left(\sigma^2, \boldsymbol{\theta}\right)$. While both approaches can be applied below, this paper will follow Schonlau (1997) and Jones et al. (1998) by using estimated parameters in the BLUP and MSPE formulas.

The EGO algorithm is based on a heuristically selected improvement function defined for each new potential input $\boldsymbol{x}$. Schonlau (1997) proposed using the theoretical improvement function

$$I(y(\boldsymbol{x})) = \left(y^n_{min} - y(\boldsymbol{x})\right) 1_{[y^n_{min} > y(\boldsymbol{x})]}, \tag{5}$$

where y^n_{min} is the smallest element in $\boldsymbol{y}^n$ and 1_E is 1 if E is true and is 0 if E is false (see also Jones et al., 1998; Huang et al., 2006). Of course, $I(y(\boldsymbol{x}))$ is unknown but a probabilistic assessment can be made of its possible values by substituting $Y(\boldsymbol{x})$ for $y(\boldsymbol{x})$. The expected improvement is defined to be conditional expectation of $I(Y(\boldsymbol{x}))$ given $\boldsymbol{y}^n$, i.e., $EI(\boldsymbol{x}) = E\{I(Y(\boldsymbol{x}))|\boldsymbol{y}^n\}$. This quantity can be shown to be approximately

$$EI(\boldsymbol{x}) = \left[(y^n_{min} - \widehat{y}(\boldsymbol{x}))\Phi\left(\frac{y^n_{min} - \widehat{y}(\boldsymbol{x})}{s(\boldsymbol{x})}\right) + s(\boldsymbol{x})\phi\left(\frac{y^n_{min} - \widehat{y}(\boldsymbol{x})}{s(\boldsymbol{x})}\right)\right] 1_{[s(\boldsymbol{x})>0]} \tag{6}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function and $\phi(\cdot)$ is the associated density function. The value of $EI(\boldsymbol{x})$ will be large if either the predicted value $\widehat{y}(\boldsymbol{x})$ is much smaller than y^n_{min} *or* $s(\boldsymbol{x})$ is large (the latter means there is a large amount of uncertainty in the estimated $y(\boldsymbol{x})$). The steps of the EGO algorithm are

1. Evaluate $y(\cdot)$ at an initial space-filling design $\mathcal{D}_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, such as a maximin Latin hypercube.
2. Estimate the stochastic process parameters β, σ^2, and $\boldsymbol{\theta}$ based on $\boldsymbol{y}^n$.
3. Find $\boldsymbol{x}^{n+1} \in \operatorname{argmax} EI(\boldsymbol{x})$.
4. Evaluate $y(\boldsymbol{x}^{n+1})$, increment n, and go to Step 2 unless a stopping criterion has been met.

This paper will generalize the philosophy of the EGO algorithm to construct a finite approximation to the Pareto Set and the Pareto Front.

3. Modeling multiple outputs using multivariate Gaussian processes

Now let $\boldsymbol{y}(\boldsymbol{x}) = (y_1(\boldsymbol{x}), \ldots, y_m(\boldsymbol{x}))$ denote an m-dimensional computer simulator output where $\boldsymbol{x}$ in $\mathcal{X}$. This paper assumes that $\boldsymbol{y}(\boldsymbol{x})$ can be modeled as a draw from an m-variate Gaussian process of the form

$$\boldsymbol{Y}(\boldsymbol{x}) = (Y_1(\boldsymbol{x}), \ldots, Y_m(\boldsymbol{x}))^\top = \boldsymbol{\beta} + \boldsymbol{A}\boldsymbol{Z}(\boldsymbol{x}) \tag{7}$$

where $\boldsymbol{A} = (a_{ij})$ is a symmetric $m \times m$ positive-definite matrix,

$$\boldsymbol{\beta} = (\beta_1 \cdots \beta_m)^\top, \tag{8}$$

and $\boldsymbol{Z}(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), \ldots, Z_m(\boldsymbol{x}))^T$ is an $m \times 1$ vector of mutually independent stationary Gaussian processes each with *zero mean* and *unit variance*, and $Z_i(\cdot)$ has correlation function of the form

$$R(\boldsymbol{x} - \boldsymbol{x}^\star \mid \boldsymbol{\theta}_i) = \exp\left\{\sum_{j=1}^{d} \theta_{i,j}\left(x_i - x_i^\star\right)^2\right\}. \tag{9}$$

Here, $\boldsymbol{x}$ and $\boldsymbol{x}^\star$ are assumed to be any arbitrary inputs in $\mathcal{X}$. It is straightforward to show that $Y_i(\cdot)$ has mean β_i and

$$Cov(\boldsymbol{Y}(\boldsymbol{x}), \boldsymbol{Y}(\boldsymbol{x}^\star)) = \boldsymbol{A}\, diag\left(R(\boldsymbol{x} - \boldsymbol{x}^\star \mid \boldsymbol{\theta}_1), \ldots, R(\boldsymbol{x} - \boldsymbol{x}^\star \mid \boldsymbol{\theta}_m)\right) \boldsymbol{A}^\top, \tag{10}$$

where *diag* $(\boldsymbol{v})$ of the $n \times 1$ vector $\boldsymbol{v}$ denotes the $n \times n$ diagonal matrix with elements $\boldsymbol{v}$. Taking $\boldsymbol{x} = \boldsymbol{x}^\star$ in (10) gives the variance–covariance matrix of $\boldsymbol{Y}(\boldsymbol{x})$,

$$Cov(\boldsymbol{Y}(\boldsymbol{x}), \boldsymbol{Y}(\boldsymbol{x})) = \boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{A}\boldsymbol{A} \equiv \Sigma_0. \tag{11}$$

Thus model (7) states that each component $Y_i(\boldsymbol{x})$ is stationary with component specific mean, variance $\sum_{j=1}^m a_{ij}^2$ while the variance–covariance matrix of $\boldsymbol{Y}(\boldsymbol{x})$ is independent of $\boldsymbol{x}$. When $\boldsymbol{A}$ is diagonal, $Y_1(\boldsymbol{x}), \ldots, Y_m(\boldsymbol{x})$ are mutually independent.

Suppose that $\boldsymbol{y}(\cdot)$ has been evaluated at the n inputs in $\mathcal{D}_n = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \subset \mathcal{X}$. Let $\boldsymbol{y}^{mn} = (\boldsymbol{y}^\top(\boldsymbol{x}_1), \ldots, \boldsymbol{y}^\top(\boldsymbol{x}_n))^\top$ denote the associated $mn \times 1$ vector of stacked $\boldsymbol{y}^\top(\boldsymbol{x}_i)$ outputs and $\boldsymbol{Y}^{mn}$ the corresponding process values. Let Σ_{mn} denote the $mn \times mn$ variance–covariance matrix of $\boldsymbol{Y}^{mn}$; it is easy to compute that Σ_{mn} is

$$\begin{pmatrix} \Sigma_0 & Cov(\boldsymbol{Y}(\boldsymbol{x}_1), \boldsymbol{Y}(\boldsymbol{x}_2)) & \cdots & Cov(\boldsymbol{Y}(\boldsymbol{x}_1), \boldsymbol{Y}(\boldsymbol{x}_n)) \\ Cov(\boldsymbol{Y}(\boldsymbol{x}_1), \boldsymbol{Y}(\boldsymbol{x}_2)) & \Sigma_0 & \cdots & Cov(\boldsymbol{Y}(\boldsymbol{x}_2), \boldsymbol{Y}(\boldsymbol{x}_n)) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(\boldsymbol{Y}(\boldsymbol{x}_1), \boldsymbol{Y}(\boldsymbol{x}_n)) & Cov(\boldsymbol{Y}(\boldsymbol{x}_2), \boldsymbol{Y}(\boldsymbol{x}_n)) & \cdots & \Sigma_0 \end{pmatrix}. \tag{12}$$

For any given input $\boldsymbol{x}_0$, the $m \times mn$ covariance of $\boldsymbol{Y}(\boldsymbol{x}_0)$ and $\boldsymbol{Y}^{mn}$ is denoted by

$$\Sigma_{0,mn} = (Cov(\boldsymbol{Y}(\boldsymbol{x}_0), \boldsymbol{Y}(\boldsymbol{x}_1)), \ldots, Cov(\boldsymbol{Y}(\boldsymbol{x}_0), \boldsymbol{Y}(\boldsymbol{x}_n))). \tag{13}$$

This paper considers two choices of model forms for $\boldsymbol{A}$. The first model form assumes that $\boldsymbol{A}$ is a *diagonal* with *positive* entries; as noted above, this assumption fits independent GPs to each $y_i(\cdot)$, $1 \le i \le m$ and is termed the *Independence* model. The second model form assumes $\boldsymbol{A}$ is a *symmetric, positive-definite* matrix. This model permits dependence *among* the various outputs; it is termed the *Nonseparable Dependence* model. The matrix $\boldsymbol{A}$ in the nonseparable dependence model can be thought of as the unique matrix square root of Σ_0 which can be calculated via eigendecomposition. While some geostatistics literature proposes treating $\boldsymbol{A}$ as the lower triangular Cholesky decomposition of Σ_0 (see Gelfand et al., 2004; Banerjee et al., 2008), Fricker et al. (2013) shows that such a specification induces artificial asymmetry into the covariance structure and therefore argues that the eigendecomposition is more appropriate for modeling functions having no *a priori* hierarchy of dependence.

When $\boldsymbol{\beta}$, $\boldsymbol{A}$, and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{\boldsymbol{m}})$ are known, the Gaussian assumption gives that

$$\left[\begin{pmatrix} \boldsymbol{Y}(\boldsymbol{x}_0) \\ \boldsymbol{Y}^{mn} \end{pmatrix} \middle| \boldsymbol{\beta} \right] \sim N\left(\begin{bmatrix} \boldsymbol{I}_m \\ \mathcal{F} \end{bmatrix} \boldsymbol{\beta}, \begin{bmatrix} \Sigma_0 & \Sigma_{0,mn} \\ \Sigma_{0,mn}^\top & \Sigma_{mn} \end{bmatrix} \right), \tag{14}$$

where $\mathcal{F} = \boldsymbol{1}_n \otimes \boldsymbol{I}_m$. Therefore, standard multivariate normal results yield

$$\left[\boldsymbol{Y}(\boldsymbol{x}_0) | \boldsymbol{Y}^{mn} = \boldsymbol{y}^{mn}, \boldsymbol{\beta} \right] \sim N\left(\boldsymbol{\beta} + \Sigma_{0,mn}\Sigma_{mn}(\boldsymbol{y}^{mn} - \mathcal{F}\boldsymbol{\beta}),\ \Sigma_0 - \Sigma_{0,mn}\Sigma_{mn}^{-1}\Sigma_{0,mn}^\top \right). \tag{15}$$

Integrating out $\boldsymbol{\beta}$ in (15) with respect to the non-informative prior $[\boldsymbol{\beta}] \propto 1$ yields

$$\left[\boldsymbol{Y}(\boldsymbol{x}_0) | \boldsymbol{Y}^{mn} = \boldsymbol{y}^{mn} \right] \sim N\left(\widehat{\boldsymbol{y}}(\boldsymbol{x}_0),\ \boldsymbol{S}(\boldsymbol{x}_0) \right), \tag{16}$$

where

$$\widehat{\boldsymbol{y}}(\boldsymbol{x}_0) = \widehat{\boldsymbol{\beta}}_{\mathrm{GLS}} + \Sigma_{0,mn}\Sigma_{mn}^{-1}(\boldsymbol{y}^{mn} - \mathcal{F}\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}}), \quad \text{with} \tag{17}$$

$$\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}} = (\mathcal{F}^\top \Sigma_{mn}^{-1}\mathcal{F})^{-1}\mathcal{F}^\top \Sigma_{mn}^{-1}\boldsymbol{y}^{mn}, \tag{18}$$

and estimated prediction uncertainty

$$\boldsymbol{S}(\boldsymbol{x}_0) = \Sigma_0 - \Sigma_{0,mn}\Sigma_{mn}^{-1}\Sigma_{0,mn}^\top + (\boldsymbol{I}_m - \Sigma_{0,mn}\Sigma_{mn}^{-1}\mathcal{F}) \times (\mathcal{F}^\top \Sigma_{mn}^{-1}\mathcal{F})^{-1} \times (\boldsymbol{I}_m - \Sigma_{0,mn}\Sigma_{mn}^{-1}\mathcal{F})^\top. \tag{19}$$

When $\boldsymbol{A}$ and $\boldsymbol{\theta}$ are unknown, this paper plugs restricted maximum likelihood (REML) estimates

$$\left(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{\theta}}\right) \in \operatorname{argmax} \left\{ -\frac{1}{2}\log\left(|\Sigma_{mn}|\right) - \frac{1}{2}\log\left(|\mathcal{F}^\top \Sigma_{mn}^{-1}\mathcal{F}|\right) - \frac{1}{2}\left(\boldsymbol{y}^{mn} - \mathcal{F}\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}}\right)^\top \Sigma_{mn}^{-1}\left(\boldsymbol{y}^{mn} - \mathcal{F}\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}}\right) \right\}. \tag{20}$$

into (17)–(19) to provide $\boldsymbol{y}(\boldsymbol{x}_0)$ estimates and uncertainty quantifications.

The most obvious advantage of the nonseparable dependence model is it allows one to incorporate more realistic assumptions regarding the between-output relationship into the Gaussian process model. Consider the hypothetical example where a computer simulator's inputs $\boldsymbol{x}$ are various design parameters for an automobile and the outputs $y_1(\boldsymbol{x})$ and $y_2(\boldsymbol{x})$ measure the fuel efficiency and acceleration of the automobile. The assumption that these outputs are negatively correlated is reasonable and consistent with real-world experience. The nonseparable dependence model allows one to capture this correlation in the output vectors, while the independence model assumes that these outputs vary independently. The most obvious disadvantage of the nonseparable dependence model is in the estimation of the covariance parameters. In the independence model, the off-diagonal elements of $\boldsymbol{A}$ are fixed at 0. Therefore, there are $m(m-1)/2$ fewer parameters to

estimate when compared to the nonseparable dependence model, making the maximization of the likelihood function a simpler optimization problem.

An extensive study of the advantages and disadvantages of the nonseparable dependence model with plug-in estimates of covariance parameters on some actual multiple-output computer simulators can be found in Fricker et al. (2013). The authors compared a simpler independence model to a variety of covariance structures, including the nonseparable covariance structure described previously in this section (referred in their paper as the nonseparable *linear model of coregionalization* or *LMC*) using two real-world case studies. Here is a summary of their key findings. The main advantage of the nonseparable dependence model was in its approximation of the distribution of scalar functions of simulator outputs. For both case studies, a domain-specific non-linear scalar function of the simulator output was introduced. The posterior distribution of a scalar function can be approximated by drawing samples from (16) and applying the function to these samples. The metric D^{α}, defined as the proportion of $100\alpha\%$ posterior credible intervals that contain the actual output of the scalar function, was computed for both the independence model and the nonseparable LMC. This value was then plotted against α for $\alpha \in [0, 1]$. A plot with large deviations from a straight line through the origin with unit slope indicates a lack of fit. For both examples' domain-specific scalar functions, the nonseparable LMC had D^{α} vs. α plots with only slight deviations from the reference line. However, the independence model had D^{α} vs. α plots with rather large deviations from the reference line. This suggests that the nonseparable LMC better quantifies the uncertainty in scalar functions of the multiple outputs of the computer simulator. The main drawback of the nonseparable LMC relative to the independence model discovered in these case studies was in regards to marginal output prediction. In both case studies, the prediction of individual outputs was shown to have roughly equal or much lower root mean square prediction error (RMSE) when using the independence model.

4. The expected maximin fitness improvement function

This section proposes an improvement function tailored to the Pareto optimization problem. Let $\mathcal{P}_{\mathcal{Y}}^{n}$ denote the set of nondominated outputs among the first n computed output vectors, and $\mathcal{P}_{\mathcal{X}}^{n}$ the associated set of $\boldsymbol{x}$ inputs; thus $\mathcal{P}_{\mathcal{X}}^{n} = \left\{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_p^*\right\}$ with $p \leq n$ and $\mathcal{P}_{\mathcal{Y}}^{n} = \left\{\boldsymbol{y}(\boldsymbol{x}_1^*), \ldots, \boldsymbol{y}(\boldsymbol{x}_p^*)\right\}$.

This paper proposes use of the *truncated maximin fitness function*

$$I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x})\right) \equiv -\max_{\mathbf{x}_i \in \mathcal{P}_{\mathcal{X}}^{n}} \min_{j=1,\ldots,m} \left(y_j(\mathbf{x}) - y_j(\mathbf{x}_i)\right) \times 1_{\left[-\max_{\mathbf{x}_i \in \mathcal{P}_{\mathcal{X}}^{n}} \min_{j=1,\ldots,m} \left(y_j(\mathbf{x}) - y_j(\mathbf{x}_i)\right) > 0\right]} \tag{21}$$

as the (theoretical) improvement that $y(\boldsymbol{x})$ makes to the current Pareto Front where the indicator function 1_E is 1 or 0 according as the event E is true or not. The effectiveness of the maximin fitness function to identify promising new inputs $\boldsymbol{x}$ depends on scales of $y_1(\cdot), \ldots, y_m(\cdot)$ being comparable. In the algorithm below, each $y_i(\cdot)$, for $i = 1, \ldots, m$, is empirically scaled using the initial set of training data to have a maximum of 1 and minimum of 0, i.e., $\min\{y_i(\boldsymbol{x}_1), \ldots, y_i(\boldsymbol{x}_n)\} = 0$ and $\max\{y_i(\boldsymbol{x}_1), \ldots, y_i(\boldsymbol{x}_n)\} = 1$.

A non-truncated version of (21) was introduced in Bautista (2009). The source of both versions was the "modified maximin fitness function"

$$\max_{\mathbf{x}_i \in \mathcal{P}_{\mathcal{X}}^{n}} \min_{j=1,\ldots,m} \left(y_j(\mathbf{x}) - y_j(\mathbf{x}_i)\right) \tag{22}$$

introduced in Balling (2003) as a component of a multiobjective evolutionary algorithm. As will be shown below, the truncated maximin fitness function can also be motivated in terms of the additive binary-ϵ measure.

Four properties of $I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x})\right)$ will be described that motivate its use as an improvement function. First, when $m = 1$, it is straightforward to show that $I_{\mathcal{M}}(y(\boldsymbol{x})) = I(y(\boldsymbol{x}))$, the theoretical EGO improvement function, so that single-objective improvement is a special case of multiobjective improvement. Second, it is easy to show that for a (candidate) $\boldsymbol{y}(\boldsymbol{x}) \notin \mathcal{P}_{\mathcal{Y}}^{n}$, $I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x})\right) > 0$ if and only if $\boldsymbol{y}(\mathbf{x})$ is not dominated by any vector in $\mathcal{P}_{\mathcal{Y}}^{n}$, and $I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x})\right) = 0$ if and only if $\boldsymbol{y}(\mathbf{x})$ is dominated by a vector in $\mathcal{P}_{\mathcal{Y}}^{n}$. (If $\boldsymbol{y}(\boldsymbol{x}) \in \mathcal{P}_{\mathcal{Y}}^{n}$ then $I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x})\right) = 0$ Balling, 2003.) Third, $I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x})\right)$ is monotonic with respect to Pareto dominance, in the sense that $I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x})\right) \geq I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x}^{\star})\right)$ provided that $\boldsymbol{y}(\mathbf{x}) \succeq \boldsymbol{y}(\mathbf{x}^{\star})$. Fourth, and last, $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$ is equivalent to the additive binary-ϵ indicator of an appropriately selected comparison of Pareto Front approximations. To describe the equivalence, start with definition of the additive binary-ϵ indicator, which is a popular Pareto set approximation quality indicator introduced in Zitzler et al. (2003) that allows one to compare two Pareto Front approximations B and C in the objective space. Specifically, the additive binary-ϵ indicator of C relative to B is the smallest real number (positive or negative) that must be added to all vectors in C (thus "worsening" C if $\epsilon > 0$) so that the set B dominates the degraded set C, i.e.,

$$I_{\epsilon^+}(B, C) = \inf_{\epsilon \in \mathbb{R}} \left\{\forall\, \boldsymbol{y}^c \in C\ \exists\, \boldsymbol{y}^b \in B : y_i^b \leq \epsilon + y_i^c\ \forall\, i = 1, \ldots, m\right\}. \tag{23}$$

In words, the additive binary-ϵ indicator of C relative to B measures how much "better" C is than B in terms of dominance.

To describe the relationship between the maximin fitness function and the additive binary-ϵ indicator, let $\mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})$ be the updated Pareto front if $\boldsymbol{y}^{mn}$ is augmented by $\boldsymbol{y}(\boldsymbol{x})$. Then, one can regard $I_{\epsilon^+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})\right)$ as quantifying how much $\boldsymbol{y}(\boldsymbol{x})$ improves upon the current Pareto front approximation $\mathcal{P}_{\mathcal{Y}}^{n}$. It is reasonable, then, to use $I_{\epsilon^+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})\right)$ as an

improvement function. The following theorem shows that this approach is equivalent to using the maximin fitness function to measure improvement.

Theorem 4.1. *Let* $\mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})$ *be the set of nondominated points in the set* $\mathcal{P}_{\mathcal{Y}}^{n} \cup \{\boldsymbol{y}(\boldsymbol{x})\}$. *Then,* $I_{\epsilon+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})\right) = I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$.

The proof of Theorem 4.1 is given in the Appendix A.

Note that the inclusion of the indicator function in (21) (the truncation) is a critical piece of three of the desirable properties described above. Without it, the truncated maximin fitness function is not a generalization of the EGO improvement function *because* it is not equal to zero when $\boldsymbol{y}(\boldsymbol{x})$ is dominated by a vector in $\mathcal{P}_{\mathcal{Y}}^{n}$. Moreover, the equality between $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$ and $I_{\epsilon+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})\right)$ no longer holds.

4.1. Expected maximin fitness improvement

While $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$ is not known, one can replace $\boldsymbol{y}(\boldsymbol{x})$ by $\boldsymbol{Y}(\boldsymbol{x})$ and use the conditional distribution of $I_{\mathcal{M}}(\boldsymbol{Y}(\mathbf{x}))$ given the current training data to assess the improvement that $\boldsymbol{x}$ adds to the current data. In particular, the conditional expected mean of $I_{\mathcal{M}}(\boldsymbol{Y}(\mathbf{x}))$ is used below to assess the improvement possible at $\boldsymbol{x}$.

The expected maximin fitness (EMmI) function is defined to be

$$EI_{\mathcal{M}}(\boldsymbol{x}) = E\{I_{\mathcal{M}}(\boldsymbol{Y}(\mathbf{x}))|\boldsymbol{Y}^{mn} = \boldsymbol{y}^{mn}\}. \tag{24}$$

The EMmI function is used to control the search for the Pareto Front. Equivalently, the previous subsection shows that $EI_{\mathcal{M}}(\boldsymbol{x})$ is the same as

$$E\left\{I_{\epsilon+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})\right)|\boldsymbol{Y}^{mn} = \boldsymbol{y}^{mn}\right\} \tag{25}$$

where $\boldsymbol{y}(\boldsymbol{x})$ in $I_{\epsilon+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})\right)$ is replaced by $\boldsymbol{Y}(\boldsymbol{x})$.

There are two cases that will be considered in describing the calculation of $EI_{\mathcal{M}}(\boldsymbol{x})$. The first case is $m = 2$ where a nearly a closed-form expression for $EI_{\mathcal{M}}(\boldsymbol{x})$ is derived and implemented in code. In the second case is $m \geq 3$ where a Monte Carlo method is used to estimate $EI_{\mathcal{M}}(\boldsymbol{x})$; details of our implementation are given in the next section.

When $m = 2$, $\boldsymbol{Y}(\boldsymbol{x}) = (Y_1(\boldsymbol{x}), Y_2(\boldsymbol{x}))$ has conditional mean and covariance

$$\widehat{\boldsymbol{y}}(\boldsymbol{x}) = \begin{bmatrix} \widehat{y}_1(\boldsymbol{x}) \\ \widehat{y}_2(\boldsymbol{x}) \end{bmatrix} \tag{26}$$

and

$$\boldsymbol{S}(\boldsymbol{x}) = \begin{bmatrix} s_1^2(\boldsymbol{x}) & \rho(\boldsymbol{x})s_1(\boldsymbol{x})s_2(\boldsymbol{x}) \\ \rho(\boldsymbol{x})s_1(\boldsymbol{x})s_2(\boldsymbol{x}) & s_2^2(\boldsymbol{x}) \end{bmatrix}, \quad \text{say,} \tag{27}$$

respectively, where $\rho(\boldsymbol{x})$ is the correlation between the two outputs.

Without loss of generality, assume that the points are labeled so that $y_1(\boldsymbol{x}_1^*) \leq \cdots \leq y_1(\boldsymbol{x}_p^*)$. As a consequence of the fact that $\mathcal{P}_{\mathcal{Y}}^{n}$ cannot contain any dominated points, it must be the case that $y_2(\boldsymbol{x}_1^*) \geq \cdots \geq y_2(\boldsymbol{x}_p^*)$. For notational convenience, let $y_1(\boldsymbol{x}_{p+1}^*) = y_2(\boldsymbol{x}_0^*) = \infty$, $k(1) = 2$, $k(2) = 1$, $h(1, j) = j - 1$, and $h(2, j) = j + 1$. It is straightforward to prove that $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$ partitions $\mathbb{R}^2$ into $2p + 1$ regions $R_{1,1}, \ldots, R_{1,p}, R_{2,1}, \ldots, R_{2,p}$ and R_D, where, for $i = 1, 2$ and $j = 1, \ldots, p$, $R_{i,j}$ is given by

$$\left\{\left(y_i, y_{k(i)}\right) : y_i \leq y_i(\boldsymbol{x}_j^*), y_{k(i)}(\boldsymbol{x}_j^*) - y_i(\boldsymbol{x}_j^*) + y_i \leq y_{k(i)} \leq y_{k(i)}(\boldsymbol{x}_{h(i,j)}^*) - y_i(\boldsymbol{x}_j^*) + y_i\right\} \tag{28}$$

and

$$R_D = \left\{(y_1, y_2) : \{(y_1, y_2)\} \prec \mathcal{P}_{\mathcal{Y}}^{n}\right\}. \tag{29}$$

Fig. 1 illustrates this decomposition for a Pareto Front containing three points. It can be seen that $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$ is equal to $y_i(\boldsymbol{x}_j^*) - y_i$ for $\boldsymbol{x} \in R_{i,j}$, for $i = 1, 2, j = 1, \ldots, p$, while $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$ is equal to 0 for $\boldsymbol{x} \in R_D$. Therefore, letting

$$Int_{i,j} = \int_{-\infty}^{y_i(\boldsymbol{x}_j^*)} \int_{y_{k(i)}(\boldsymbol{x}_j^*) - y_i(\boldsymbol{x}_j^*) + y_i}^{y_{k(i)}(\boldsymbol{x}_{h(i,j)}^*) - y_i(\boldsymbol{x}_j^*) + y_i} \left[y_i(\boldsymbol{x}_j^*) - y_i\right] f(y_1, y_2) dy_{k(i)} dy_i \tag{30}$$

where $i = 1, 2, j = 1, \ldots, p$, and $f(y_1, y_2)$ is the bivariate conditional normal probability density function with mean (26) and covariance (27), gives

$$EI_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{2} \sum_{j}^{p} Int_{i,j}. \tag{31}$$

Finally, accounting for the different upper and lower bounds for each $Int_{i,j}$, we arrive at the following formula for $EI_{\mathcal{M}}(\boldsymbol{x})$.

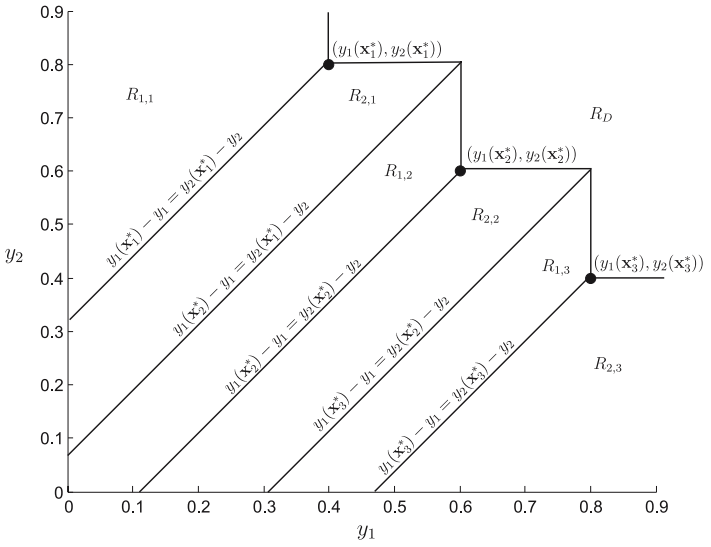

Fig. 1. Regions of integration $R_{1,1}, \ldots, R_{1,p}, R_{2,1}, \ldots, R_{2,p}$ and R_D for a $p = 3$ point Pareto Front.

Theorem 4.2. *When $m = 2$,*

$$EI_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i=1}^{2}\sum_{j=1}^{p}\left(Int_{i,j}^{1}(\boldsymbol{x}) + Int_{i,j}^{2}(\boldsymbol{x}) + Int_{i,j}^{3}(\boldsymbol{x})\right) \tag{32}$$

where

$$\begin{aligned}
Int_{i,j}^{1}(\boldsymbol{x}) \;=\; & s_i(\boldsymbol{x})\phi\left(\frac{d(i,j)}{s_i(\boldsymbol{x})}\right) \times \left[\Phi\left(\frac{-d(k(i),j) + \rho(\boldsymbol{x})s_{k(i)}(\boldsymbol{x})d(i,j)/s_i(\boldsymbol{x})}{\sqrt{(1-\rho^2(\boldsymbol{x}))s_{k(i)}^2(\boldsymbol{x})}}\right)\right. \\
& \left. - \Phi\left(\frac{-d(k(i),h(i,j)) + \rho(\boldsymbol{x})s_{k(i)}(\boldsymbol{x})d(i,j)/s_i(\boldsymbol{x})}{\sqrt{(1-\rho^2(\boldsymbol{x}))s_{k(i)}^2(\boldsymbol{x})}}\right)\right], \\
Int_{i,j}^{2}(\boldsymbol{x}) \;=\; & \sqrt{q(i,j)}\,\frac{s_i(\boldsymbol{x}) - s_{k(i)}(\boldsymbol{x})\rho(\boldsymbol{x})}{\sqrt{2\pi\,(1-\rho^2(\boldsymbol{x}))s_{k(i)}^2(\boldsymbol{x})}} \\
& \times \left[\exp\left\{-\frac{1}{2}\left[\frac{\widehat{y}_i^2(\boldsymbol{x})}{s_i^2(\boldsymbol{x})} + \frac{\left(y_i(\boldsymbol{x}_j^*) - d(k(i),j) + \rho(\boldsymbol{x})s_{k(i)}(\boldsymbol{x})\widehat{y}_i(\boldsymbol{x})/s_i(\boldsymbol{x})\right)^2}{\sqrt{(1-\rho^2(\boldsymbol{x}))s_{k(i)}^2(\boldsymbol{x})}}\right]\right\}\right. \\
& \times \exp\left\{\frac{1}{2}q(i,j)v^2(i,j)\right\}\Phi\left(\frac{y_i(\boldsymbol{x}_j^*) - q(i,j)v(i,j)}{\sqrt{q(i,j)}}\right) \\
& \left. - \exp\left\{-\frac{1}{2}\left[\frac{\widehat{y}_i^2(\boldsymbol{x})}{s_i^2(\boldsymbol{x})} + \frac{\left(y_i(\boldsymbol{x}_j^*) - d(k(i),h(i,j)) + \rho(\boldsymbol{x})s_{k(i)}(\boldsymbol{x})\widehat{y}_i(\boldsymbol{x})/s_i(\boldsymbol{x})\right)^2}{\sqrt{(1-\rho^2(\boldsymbol{x}))s_{k(i)}^2(\boldsymbol{x})}}\right]\right\}\right. \\
& \left. \times \exp\left\{\frac{1}{2}q(i,j)v^2(i,h(i,j))\right\}\Phi\left(\frac{y_i(\boldsymbol{x}_j^*) - q(i,j)v(i,h(i,j))}{\sqrt{q(i,j)}}\right)\right]
\end{aligned}$$

and

$$\begin{aligned}
Int_{i,j}^{3}(\boldsymbol{x}) \;=\; & \left(y_i(\boldsymbol{x}_j^*) - \widehat{y}_i(\boldsymbol{x})\right) \times \left[\int_0^{u(i,j)} \Phi\left(\frac{d(i,j) - d(k(i),j) + (s_{k(i)}(\boldsymbol{x})\rho(\boldsymbol{x}) - s_i(\boldsymbol{x}))\Phi^{-1}(w)}{\sqrt{(1-\rho^2(\boldsymbol{x}))s_{k(i)}^2(\boldsymbol{x})}}\right) dw\right. \\
& \left. - \int_0^{u(i,j)} \Phi\left(\frac{d(i,j) - d(k(i),h(i,j)) + (s_{k(i)}(\boldsymbol{x})\rho(\boldsymbol{x}) - s_i(\boldsymbol{x}))\Phi^{-1}(w)}{\sqrt{(1-\rho^2(\boldsymbol{x}))s_{k(i)}^2(\boldsymbol{x})}}\right) dw\right]
\end{aligned}$$

with constants

$$u(i,j) = \Phi\left(\frac{d(i,j)}{s_i(\boldsymbol{x})}\right)$$
$$v(i,j) = \frac{\widehat{y}_i(\boldsymbol{x})}{s_i^2(\boldsymbol{x})} + \frac{y_2(\boldsymbol{x}_j^*) - d(k(i),j) + \rho(\boldsymbol{x})s_{k(i)}(\boldsymbol{x})\widehat{y}_i(\boldsymbol{x})/s_i(\boldsymbol{x})}{\sqrt{(1-\rho^2(\boldsymbol{x}))s_{k(i)}^2(\boldsymbol{x})}}$$
$$q(i,j) = \frac{(1-\rho^2(\boldsymbol{x}))s_{k(i)}^2(\boldsymbol{x})s_i^2(\boldsymbol{x})}{s_i^2(\boldsymbol{x}) + s_{k(i)}^2(\boldsymbol{x}) - 2\rho(\boldsymbol{x})s_i(\boldsymbol{x})s_{k(i)}(\boldsymbol{x})}$$
$$d(i,j) = y_i(\boldsymbol{x}_j^*) - \widehat{y}_i(\boldsymbol{x}).$$

The Supplementary material provides details of this formula by showing that $Int_{i,j} = Int_{i,j}^1 + Int_{i,j}^2 + Int_{i,j}^3$, for $i = 1, 2, j = 1, \ldots, p$ (see Appendix B).

5. The EMmI algorithm for approximating the Pareto front and set

First, an outline of the proposed multiobjective EMmI optimization algorithm based on the expected maximin fitness function will be stated. Then some of the computational details required to implement the procedure will be discussed.

1. Evaluate $\boldsymbol{y}(\cdot)$ at an initial space-filling design $\mathcal{D}_n = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \subset \mathcal{X}$. Let $\boldsymbol{y}^{mn} = (\boldsymbol{y}^\top(\boldsymbol{x}_1), \ldots, \boldsymbol{y}^\top(\boldsymbol{x}_n))^\top$. Empirically scale the outputs so that $\min\{y_i(\boldsymbol{x}_1), \ldots, y_i(\boldsymbol{x}_n)\} = 0$ and $\max\{y_i(\boldsymbol{x}_1), \ldots, y_i(\boldsymbol{x}_n)\} = 1$.
2. Estimate $\boldsymbol{\theta}$ and $\boldsymbol{A}$ using REML based on the $\boldsymbol{y}^{mn}$ (or another method such as maximum likelihood).
3. Calculate the current Pareto Set $\mathcal{P}_{\mathcal{X}}^n$ and Pareto Front $\mathcal{P}_{\mathcal{Y}}^n$.
4. Find $\boldsymbol{x}^{n+1} \in \arg\max\ EI_{\mathcal{M}}(\boldsymbol{x})$.
5. Evaluate $\boldsymbol{y}(\boldsymbol{x}^{n+1})$. Repeat Steps 2–5 with output data

$$\boldsymbol{y}^{m(n+1)} = (\boldsymbol{y}^\top(\boldsymbol{x}_1), \ldots, \boldsymbol{y}^\top(\boldsymbol{x}_n), \boldsymbol{y}^\top(\boldsymbol{x}_{n+1}))^\top \tag{33}$$

until the computational budget has been exhausted or other stopping criteria met.

While the computational budget will often be the reason for stopping output evaluations, there are other possible stopping criteria. Perhaps the most obvious criterion is to determine the maximum expected improvement, $\max\ EI_{\mathcal{M}}(\boldsymbol{x})$ and terminate sampling if this quantity is sufficiently small (see Schonlau, 1997). Alternatively, because the correlation parameters are re-estimated after each new run, the sequence of maximum expected improvements need not be monotone decreasing; hence a stopping criterion based on having a sufficiently small maximum expected improvement after a sequence of, say 5, addition inputs are specified, is often used as a more cautious stopping criterion (Williams et al., 2000).

Four implementation choices required to implement the EMmI algorithm in the examples below will be described. First, the initial space-filling design was taken to be a maximin LHD. For many cases, users can find a mathematically provable maximin LHD from the on-line collection found at http://www.spacefillingdesigns.nl/. We used approximate maximin LHDs obtained from applying the genetic algorithm `bestlh` available in on-line supplementary material for Forrester et al. (2008) (see Appendix B).

Second, for the case of the independence model, REML estimates of $\boldsymbol{\theta}$ and the process variances were obtained using MATLAB function `mperk`, which can be obtained by contacting the second author. For the nonseparable dependence model, the MATLAB function ga, also available as a component of the on-line supplementary material to Forrester et al. (2008) (see Appendix B), was used to obtain the initial estimates of $\boldsymbol{\theta}$ and $\boldsymbol{A}$; these values were taken to be the starting points in an application of the MATLAB function `fmincon` to produce the final estimates of $\boldsymbol{\theta}$ and $\boldsymbol{A}$. In both models, the estimated model parameters were used to calculate $\boldsymbol{y}(\boldsymbol{x})$, and $\boldsymbol{S}(\boldsymbol{x})$.

Third, the MATLAB function `paretoset.m` (written by Y. Cao and available at http://www.mathworks.com/matlabcentral/fileexchange/15181-pareto-set) was used to calculate $\mathcal{P}_{\mathcal{X}}^n$ and $\mathcal{P}_{\mathcal{Y}}^n$.

Fourth, the MATLAB function `NOMADm`, Mark Abramson's MATLAB implementation of a mesh adaptive direct search (MADS) algorithm, was used to optimize $EI_{\mathcal{M}}(\boldsymbol{x})$ (see Audet and Dennis, 2006 and the URL http://www.gerad.ca/NOMAD/Abramson/nomadm.html). To maximize $EI_{\mathcal{M}}(\boldsymbol{x})$ when $m = 2$, formula 4.2 for $EI_{\mathcal{M}}(\boldsymbol{x})$ was optimized using `NOMADm`. When $m \geq 3$, $EI_{\mathcal{M}}(\boldsymbol{x})$ was optimized via sample average approximation (SAA, described in Shapiro (2003)). The idea of SSA is to construct an approximation to $EI_{\mathcal{M}}(\boldsymbol{x})$ based on a random sample from the conditional distribution of $\boldsymbol{Y}(\boldsymbol{x})$ given the current data; then this easy-to-calculate approximation is optimized. In detail, first an independent, identically distributed sample $\boldsymbol{Z}^1, \ldots, \boldsymbol{Z}^S$ were generated from a $N(\boldsymbol{1}_m, \mathrm{I}_m)$ distribution. For any given $\boldsymbol{x}$, letting $\boldsymbol{C}(\boldsymbol{x})$ be the Cholesky decomposition of $\boldsymbol{S}(\boldsymbol{x})$; each $\boldsymbol{Z}^i$ is transformed into a random variable $\boldsymbol{Y}^i(\boldsymbol{x}) = \boldsymbol{C}(\boldsymbol{X})\boldsymbol{Z}^i + \widehat{\boldsymbol{y}}(\boldsymbol{x}) \sim N(\widehat{\boldsymbol{y}}(\boldsymbol{x}), \boldsymbol{S}(\boldsymbol{x}))$. Thus, $\boldsymbol{Y}^1(\boldsymbol{x}), \ldots, \boldsymbol{Y}^S(\boldsymbol{X})$ is a sample from the conditional distribution of $\boldsymbol{Y}(\boldsymbol{x})$ given the data. The *sample average function* $\widehat{EI_{\mathcal{M}}}(\boldsymbol{x}) = \frac{1}{S}\sum_{s=1}^S I_{\mathcal{M}}^s(\boldsymbol{x})$ is a

deterministic function for a particular realization of the random sample $\boldsymbol{Z}^1, \ldots, \boldsymbol{Z}^S$) where

$$I^s_{\mathcal{M}}(\mathbf{x}) = -\max_{\mathbf{x}_i \in \mathcal{P}^n_{\mathcal{X}}} \min_{j=1,\ldots,m} \left(Y^s_j(\mathbf{x}) - y_j(\mathbf{x}_i)\right) \times 1_{\left[-\max_{\mathbf{x}_i \in \mathcal{P}^n_{\mathcal{X}}} \min_{j=1,\ldots,m} \left(Y^s_j(\mathbf{x}) - y_j(\mathbf{x}_i)\right) > 0\right]}. \tag{34}$$

The next input is found by calculating $\boldsymbol{x}^{n+1} \in \arg\max \widehat{EI_{\mathcal{M}}}(\boldsymbol{x})$ via the `NOMADm` algorithm.

6. Examples

The performance of the EMmI algorithm will be compared with that of three algorithms that replace Step 4 of the EMmI algorithm by competing improvement criterion. The first competing criterion, from Keane (2006), chooses $\boldsymbol{x}^{n+1}$ to maximize the conditional probability that $\boldsymbol{Y}(\boldsymbol{x})$ is not dominated by the current Pareto front estimate, given the first n evaluations of $\boldsymbol{y}(\cdot)$, i.e., to maximize

$$I_{PI}(\boldsymbol{x}) = P\left\{\boldsymbol{Y}(\boldsymbol{x}) \not\succeq \boldsymbol{y} \text{ for all } \boldsymbol{y} \in \mathcal{P}^n_{\mathcal{Y}} | \boldsymbol{Y}^{mn} = \boldsymbol{y}^{mn}\right\}. \tag{35}$$

Eq. (35) is termed the probability improvement (PI). The advantage of this criterion is that it is *not* dependent on the scaling of the output. Notice that (35) can also be viewed as the conditional expectation of the indicator function

$$1_{\left[\boldsymbol{y}(\boldsymbol{x}) \not\succeq \boldsymbol{y} \text{ for all } \boldsymbol{y} \in \mathcal{P}^n_{\mathcal{Y}}\right]}. \tag{36}$$

Similar to $I_{\mathcal{M}}(\boldsymbol{y}(\mathbf{x}))$, (36) is monotonic with respect to Pareto dominance, and it is a non-negative function that is only positive for $\boldsymbol{y}(\boldsymbol{x})$ not dominated by $\mathcal{P}^n_{\mathcal{Y}}$. Unlike $I_{\mathcal{M}}(\boldsymbol{y}(\mathbf{x}))$, (36) is not equivalent to $I(y(\boldsymbol{x}))$ when $m = 1$.

A second criterion, also proposed in Keane (2006) and advocated in Forrester et al. (2008), chooses $\boldsymbol{x}^{n+1}$ to maximize a centroid weighted version of the PI criterion (35), i.e., to maximize

$$I_{CWPI}(\boldsymbol{x}) = P\left\{\boldsymbol{Y}(\boldsymbol{x}) \not\succeq \boldsymbol{y} \text{ for all } \boldsymbol{y} \in \mathcal{P}^n_{\mathcal{Y}} | \boldsymbol{Y}^{mn} = \boldsymbol{y}^{mn}\right\} \times \min_{\mathbf{x}_i \in \mathcal{P}^n_{\mathcal{X}}} \sqrt{\sum_{k=1}^{m} \left(\overline{Y}_k(\mathbf{x}) - y_k(\mathbf{x}_i)\right)^2} \tag{37}$$

where $\overline{\boldsymbol{Y}}(\mathbf{x})$ is the centroid of the n outputs; $\overline{\boldsymbol{Y}}(\mathbf{x})$ is defined to be the ratio

$$\overline{\boldsymbol{Y}}(\mathbf{x}) = \frac{E\left\{\boldsymbol{Y}(\boldsymbol{x}) 1_{\left[\boldsymbol{Y}(\boldsymbol{x}) \not\succeq \boldsymbol{y} \text{ for all } \boldsymbol{y} \in \mathcal{P}^n_{\mathcal{Y}}\right)} | \boldsymbol{Y}^{mn} = \boldsymbol{y}^{mn}\right\}}{P\left\{\boldsymbol{Y}(\boldsymbol{x}) \not\succeq \boldsymbol{y} \text{ for all } \boldsymbol{y} \in \mathcal{P}^n_{\mathcal{Y}} | \boldsymbol{Y}^{mn} = \boldsymbol{y}^{mn}\right\}}.$$

Maximizing $I_{CWPI}(\boldsymbol{x})$ is called the CWPI criterion. As for the EMmI criterion, relative scaling of the various objectives must be performed when implementing the CWPI criterion. Also, the use of the CWPI criterion can be shown to be a generalization of the single-objective EGO expected improvement function. This generalization appears to be the main motivation behind the CWPI criterion.

The third criterion has already been discussed in 4. It removes the indicator function from (21). So, the improvement criteria becomes

$$EI_{\mathcal{P}}(\boldsymbol{y}(\mathbf{x})) \equiv E\{-\max_{\mathbf{x}_i \in \mathcal{P}^n_{\mathcal{X}}} \min_{j=1,\ldots,m} \left(y_j(\mathbf{x}) - y_j(\mathbf{x}_i)\right) | \boldsymbol{Y}^{mn} = \boldsymbol{y}^{mn}\}. \tag{38}$$

An algorithm using this improvement criteria was proposed in Bautista (2009) and was referred to as the *EmaX* algorithm.

In addition to their visual fit, this section will use two real-valued quantities to summarize the quality of the Pareto Front produced by the competing criterion. The two methods are the *additive binary-ϵ indicator* and the *hypervolume indicator*. The former was described in Section 4. The following paragraph gives a brief description of the latter (see Emmerich et al., 2006 for a more complete description). Zitzler et al. (2008) gives an in-depth discussion of Pareto set approximation quality indicators.

The hypervolume indicator of a Pareto Front approximation is the area (or volume) of the region dominated by the approximation relative to a fixed reference point. The hypervolume indicator of a *finite* set $\boldsymbol{B}$ which is a Pareto Front approximation relative to the reference point $\boldsymbol{r}$ is defined to be

$$I_H(\boldsymbol{B}, \boldsymbol{r}) = \int_{\mathbb{R}^m} 1_{\{\boldsymbol{y} | \boldsymbol{y} \succeq \boldsymbol{r},\ \boldsymbol{B} \succeq \{\boldsymbol{y}\}\}} d\boldsymbol{y}. \tag{39}$$

In words, $I_H(\boldsymbol{B}, \boldsymbol{r})$ is the volume of the set of points $\boldsymbol{y}$ in the objective space that dominate $\boldsymbol{r}$ and which are dominated by one or more points in $\boldsymbol{B}$ so that the larger $I_H(\boldsymbol{B}, \boldsymbol{r})$, the better the approximating set $\boldsymbol{B}$. Fig. 2 illustrates $I_H(\boldsymbol{B}, \boldsymbol{r})$ as the shaded area for an $m = 2$ dimensional example with a five point $\boldsymbol{B}$ where $\boldsymbol{r}$ is the upper right-hand corner of the shaded area.

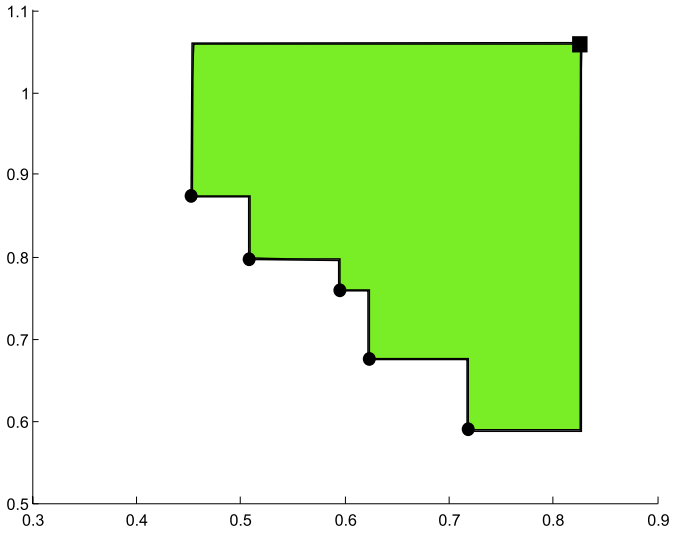

Fig. 2. The filled circles are five-point set $\boldsymbol{B}$, the filled square is the reference point $\boldsymbol{r}$, and the shaded region is $I_H(\boldsymbol{B}, \boldsymbol{r})$.

6.1. The MOP2 problem

The MOP2 test problem was first described in Fonseca and Fleming (1995). MOP2 has a $d = 2$-dimensional input space $\mathcal{X} = [-2, 2]^2$, and $m = 2$ objective functions which are

$$y_1(\boldsymbol{x}) = 1 - \exp\left\{-\sum_{i=1}^{2}\left(x_i - \frac{1}{\sqrt{2}}\right)^2\right\} \quad \text{and} \tag{40}$$

$$y_2(\boldsymbol{x}) = 1 - \exp\left\{-\sum_{i=1}^{2}\left(x_i + \frac{1}{\sqrt{2}}\right)^2\right\}. \tag{41}$$

The Pareto set is known to be the line segment

$$\mathcal{P}_{\mathcal{X}} = \left\{\boldsymbol{x} : x_1 = x_2 \text{ and } -\frac{1}{\sqrt{2}} \le x_1 \le \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \le x_2 \le \frac{1}{\sqrt{2}}\right\}. \tag{42}$$

A discrete approximation to $\mathcal{P}_{\mathcal{Y}}$ was determined by evaluating $(y_1(\boldsymbol{x}), y_2(\boldsymbol{x}))$ at 201 $\boldsymbol{x}$ points uniformly spread in $\mathcal{P}_{\mathcal{X}}$. This close approximation to $\mathcal{P}_{\mathcal{Y}}$ served as the basis for comparing the various Pareto Front approximations constructed for this example.

An initial 10 point (5 per input dimension) maximin Latin hypercube design was determined using the MATLAB function `bestlh` from Forrester et al. (2008). The initial design was augmented sequentially with 10 new inputs using EMmI and the two competing methods sketched above. Thus a total budget of 20 runs was used for this problem.

In all four cases both the independence GP model and the nonseparable dependence GP model (introduced in Section 3) were used in the conditional probability calculation. While CWPI and PI methods can be implemented using the code provided in Forrester et al. (2008), this example utilizes code written by the authors because it provides better results for PI and CWPI in terms of the hypervolume and additive binary-ϵ indicators.

To compare the Pareto Front approximations, both graphical and the Pareto set approximation quality indicators were employed. The true Pareto Front and competing Pareto Front approximations were plotted to allow visual inspection of the approximations; the spread of the approximation and its closeness to the true front were examined. The value of $I_{\epsilon^+}(\mathcal{P}_{\mathcal{Y}}, \mathcal{P}_{\mathcal{Y}}^{20})$ was calculated for each approximation, where $\mathcal{P}_{\mathcal{Y}}^{20}$ denotes the Pareto Front based on all 20 observations. Smaller values represent better approximations to the true Pareto Front. The hypervolume indicator of the various approximations was computed using $\boldsymbol{r} = (1, 1)$ as the reference point; larger values of the hypervolume indicator represent better approximations. While all of the expected improvement algorithms are deterministic, in principal, they all use maximization algorithms with stochastic search components. Therefore, these quality indicators are random variables in practice. Hence each algorithm was run five times and the mean, range, and standard deviations of the Pareto set approximation quality indicators were computed.

Table 1 shows that EMmI and EmaX, calculated using either the independence or dependence GP model for $(Y_1(\boldsymbol{x}), Y_2(\boldsymbol{x}))$, performed significantly better than either the CWPI and PI implementations, again using either the dependent or

Table 1
Summary of quality indicators in five runs of each algorithm for the *MOP2* problem.

Method	$I_{\epsilon+}(\mathcal{P}_y, \mathcal{P}_y^{10})$			$I_H(\mathcal{P}_y^{10})$		
	Mean	Range	Std Dev	Mean	Range	Std Dev
EMmI/Ind	0.0706	0.0705–0.0707	0.0001	0.2886	0.2883–0.2890	0.0002
Emax/Ind	0.0721	0.0680–0.0788	0.0043	0.2884	0.2862–0.2895	0.0013
PI/Ind	0.1368	0.0937–0.2334	0.0552	0.2531	0.2420–0.2638	0.0096
CWPI/Ind	0.0862	0.0668–0.0927	0.0112	0.2710	0.2649–0.2789	0.0060
EMmI/Dep	0.0770	0.0715–0.0882	0.0067	0.2851	0.2811–0.2889	0.0037
Emax/Dep	0.0740	0.0689–0.0797	0.0041	0.2860	0.2792–0.2894	0.0041
PI/Dep	0.1229	0.0978–0.1608	0.0256	0.2529	0.2306–0.2772	0.0226
CWPI/Dep	0.0937	0.0879–0.0977	0.0041	0.2609	0.2570–0.2647	0.0028

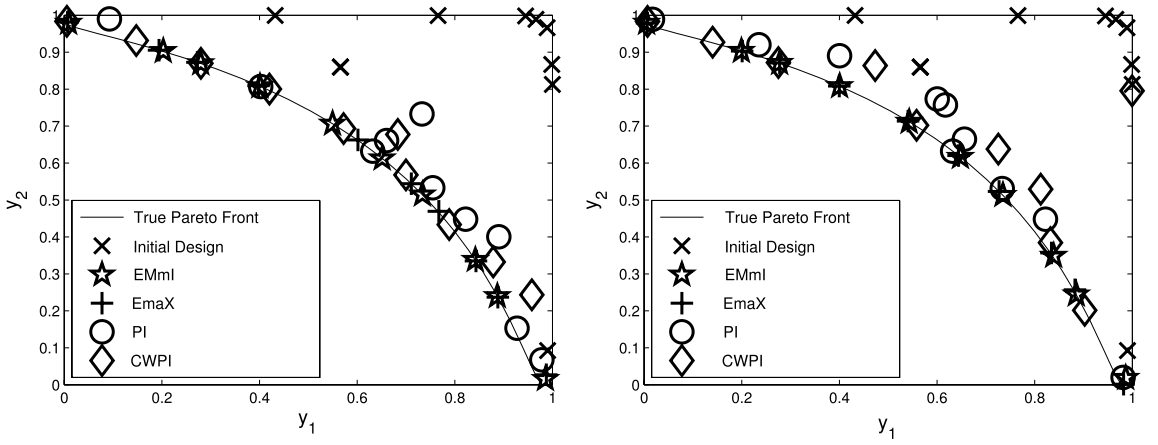

Fig. 3. Sequentially added points using an independence model (left) and dependence model (right) with EMmI (stars), EmaX(plus signs), PI (circles), and CWPI (diamonds). The initial 10 outputs are denoted by *x*'s. The smooth curve running from the top left to the bottom right of each plot is the true Pareto front.

independence GP model. The area of the dominated hypervolume when using EMmI and EmaX is *above* 0.28 on average using both the independence and dependence models, while CWPI and PI are *below* 0.28 and 0.26 on average, respectively. The additive binary-ϵ indicator is, on average, larger for CWPI and PI than for EMmI and EmaX when using either the independence and dependence model. It should also be noted that CWPI appears to outperform PI, regardless of the dependence model assumed, which is consistent with the results in Keane (2006). The plots in Fig. 3 show the results for one of the five runs; the spread and uniformity of these points support the superiority of EMmI and EmaX as well as the numerical measures. There is very little difference in performance between EMmI and EmaX. EMmI has a slight edge in the independence case (slightly lower additive binary-ϵ indicator and slightly higher hypervolume indicator), while EmaX has a slight edge in the dependence case. These differences are quite small, and neither visualization of the estimated Pareto fronts appears to be more desirable than the other. While CWPI and PI do not perform poorly, they do not appear as efficient as EMmI and EmaX because both methods have sequentially added points that are not on but only near the true Pareto front. The PI criterion appears to suffer from some clustering issues, because under both the independence and dependence GP models it has a tendency to sequentially add inputs with similar outputs, while CWPI criterion appears to be more effective at spreading out the sequentially added evaluations of the objective function.

A somewhat surprising result is that the dependence GP model appears to offer little advantage over the (less computationally demanding) independence model. On average, the dependence model performed slightly worse for almost all improvement criteria, with the single exception of the binary-ϵ indicator for the PI criterion, where slightly smaller binary-ϵ values are produced using the dependence model. One possible explanation is that the selected dependence GP model does not model pair this particular pair of functions. Other possible explanations for the inferior performance of the dependence model is that, while it is appropriate, there is too little data to reliably estimate the seven covariance parameters of the model or the maximization algorithm of the restricted likelihood function did not yield the global maximum. In contrast, the independence model requires maximization of two separate restricted likelihood functions, each of which depends on two parameters. In examples where the data was constructed to satisfy the nonseparable dependence model (see the Supplementary material, Appendix B), use of the EMmI algorithm with the dependence model produced larger hypervolume indicator and smaller binary-ϵ comparisons with the true Pareto Front than when the EMmI algorithm is used with the independence model.

Table 2
Summary of the quality indicators for five runs of each algorithm for the *DTLZ2* problem.

Method	$I_{\epsilon+}(\mathcal{P}_y, \mathcal{P}_y^{40})$			$I_H(\mathcal{P}_y^{40})$		
	Mean	Range	Std Dev	Mean	Range	Std Dev
EMmI/Ind	0.2436	0.2329–0.2519	0.0077	0.7381	0.7308–0.7447	0.0059
EmaX/Ind	0.2402	0.2347–0.2442	0.0040	0.7300	0.7265–0.7344	0.0035
CWPI/Ind	0.3023	0.2557–0.3324	0.0317	0.6684	0.6323–0.7120	0.0292
PI/Ind	0.4294	0.3675–0.4476	0.0345	0.5968	0.5738–0.6261	0.0215
EMMI/Dep	0.3044	0.2925–0.3221	0.0117	0.6960	0.6684–0.7130	0.0173
EmaX/Dep	0.2853	0.2743–0.2995	0.0094	0.7106	0.7001–0.7239	0.0105
CWPI/Dep	0.2980	0.2762–0.3192	0.0178	0.6894	0.6445–0.7193	0.0278
PI/Dep	0.3926	0.2838–0.4435	0.0681	0.6273	0.5887–0.6563	0.0272

6.2. DTLZ2 function

This example evaluates the performance of the various methods in a higher-dimensional case. To do so, the DTLZ2 test function, described in Deb et al. (2005), is used. DTLZ2 was designed to be scalable in both the number of inputs and outputs. This example considers the case where there are $m = 4$ outputs and $d = 4$ inputs. The input space is $\mathcal{X} = [0, 1]^4$. The outputs are

$$y_1(\boldsymbol{x}) = (1 + g(x_4)) \cos\left(\frac{\pi x_1}{2}\right) \cos\left(\frac{\pi x_2}{2}\right) \cos\left(\frac{\pi x_3}{2}\right) \tag{43}$$

$$y_2(\boldsymbol{x}) = (1 + g(x_4)) \sin\left(\frac{\pi x_3}{2}\right) \cos\left(\frac{\pi x_1}{2}\right) \cos\left(\frac{\pi x_2}{2}\right) \tag{44}$$

$$y_3(\boldsymbol{x}) = (1 + g(x_4)) \sin\left(\frac{\pi x_2}{2}\right) \cos\left(\frac{\pi x_1}{2}\right) \tag{45}$$

$$y_4(\boldsymbol{x}) = (1 + g(x_4)) \sin\left(\frac{\pi x_1}{2}\right) \tag{46}$$

where

$$g(x_4) = (x_4 - 0.5)^2 . \tag{47}$$

The Pareto set is $\mathcal{P}_{\mathcal{X}} = \{\boldsymbol{x} : x_4 = 0.5\}$ and $\mathcal{P}_y$ is the concave set where $g(x_4) = 0$. A discrete approximation to $\mathcal{P}_y$ was created by evaluating DTLZ2 at 20,000 points uniformly spread in $\mathcal{P}_{\mathcal{X}}$.

Proceeding in a similar fashion as the MOP2 example, an initial 20 point maximin LHD was constructed using the MATLAB function `bestlh` from Forrester et al. (2008). Then the original design was augmented sequentially with 20 new points chosen by the EMmI, CWPI, and PI algorithms using both the independence and dependence GP models. Thus a total budget of 40 evaluations of each objective function was used for this problem.

In this $m = 4$ example, graphical methods are problematic to interpret; thus only the hypervolume indicator $I_H(\mathcal{P}_y^{40})$ and the additive binary-ϵ indicator $I_{\epsilon+}(\mathcal{P}_y, \mathcal{P}_y^{40})$ will be used here to compare the algorithms. As in the previous example, each algorithm/process model was run five times and the mean, range, and standard deviation of the two comparison measures are reported in Table 2.

The DTLZ2 results based on the independence GP model were similar to the MOP2 results. In every run, EMmI/Ind and EmaX/Ind outperformed CWPI/Ind and CWPI/Ind outperformed PI–Ind in terms of both the binary-ϵ indicator and hypervolume. These quality indicators were nearly identical, on average, for EMmI/Ind and EmaX/Ind. For the dependence GP model, the results differed from those of the MOP2 example. EmaX/Dep was the clear winner among the dependence models, as it had the smallest average binary-ϵ indicator and largest average hypervolume. EMmI/Dep had a slightly larger hypervolume indicator than CWPI/Dep on average, but CWPI/Dep had a slightly smaller binary-ϵ indicator than EMmI/Dep on average. The range of both performance measures showed considerable overlap between the two improvement criteria. PI/Dep is still performed considerably worse than both EMmI/Dep and CWPI/Dep. and CWPI outperforms PI in terms of both performance measures.

The higher dimensional DTLZ2 example provided some evidence of the usefulness of the nonseparable dependence GP model. While EMmI/Dep and EmaX/Dep perform considerably poorer with the dependence model in terms of the two Pareto set quality measures, both CWPI/Dep and PI/Dep did seem to have, on average, slightly better performance when using the dependence model. The major downside of the dependence model in this example was that it depends on 26 parameters which must be estimated. This is much more difficult than the optimization problem posed by the independence model, which only requires maximization of four separate restricted likelihood functions, each of which depends on five parameters.

7. Conclusions and discussion

This paper introduces a sequential design for a computer experiment involving $m \geq 2$ expensive-to-evaluate computer simulators to approximate their Pareto Front and Pareto Set. The design uses an expected improvement algorithm based on

an interpolating stochastic process. Two versions of the algorithm are implemented: the first uses independent processes to model each output and the second uses a multivariate process that allows dependence among the outputs. The latter was considered to potentially provide additional predictive accuracy in applications where knowledge of the value of one output at the current set of input data provides information about the value of a different output at "nearby" inputs.

A closed-form expression is given for the proposed expected improvement function when $m = 2$; a Monte-Carlo approximation to the expected improvement function is presented when $m \geq 3$. Several desirable properties of the proposed improvement function are shown. Based on the examples presented in the paper and additional ones that are given in the Supplementary material (see Appendix B), the authors recommend using the expected maximin improvement computed using independent Gaussian process models (EMmI/Ind), particularly for problems where it is not possible to supply information concerning possible dependences among the output functions and where scaling of the objectives can be roughly determined. This combination of improvement function and dependence structure outperformed PI and CWPI in both the independence and dependence case. While the performance of EmaX/Ind is nearly identical to EMmI/Ind on the test problems, one can argue that EMmI/Ind is preferable because it is a proper generalization of the Efficient Global Optimization (EGO) algorithm and because of its equivalence to an improvement function based on the additive binary-ϵ indicator.

Another advantage of the independence model is that it can more easily handle cases where m is large. Not only are there fewer parameters to estimate, but the optimization of the likelihood function can be done separately for each output rather than jointly. Meanwhile, for larger m, optimization of the full nonseparable dependence model can become cumbersome and numerical issues can arise. Therefore, for large m, it is recommended that the independence model be used.

We mention one alternative criterion to EMmI/Ind that has attractive performance although it is more difficult to implement than EMmI/Ind. This criterion is based on the hypervolume indicator defined in (39). To describe a "hypervolume improvement function" in the spirit of Section 5, fix an upper bound $\boldsymbol{r}$ for the vector of output functions and define

$$I_{\mathcal{H}}(\mathbf{y}(\boldsymbol{x})) = \begin{cases} 0, & \text{if } \boldsymbol{y}(\boldsymbol{x}) \preceq \mathcal{P}_{\mathcal{Y}}^{n} \quad \text{or} \quad \boldsymbol{y}(\boldsymbol{x}) \not\succeq \boldsymbol{r} \\ I_H\left(\{\boldsymbol{y}(\boldsymbol{x})\} \cup \mathcal{P}_{\mathcal{Y}}^{n}, \boldsymbol{r}\right) - I_H\left(\mathcal{P}_{\mathcal{Y}}^{n}, \boldsymbol{r}\right), & \text{otherwise.} \end{cases} \tag{48}$$

The corresponding update function selects $\boldsymbol{x}^{n+1}$ to maximize the *expected hypervolume improvement*

$$EI_{\mathcal{H}}(\boldsymbol{x}) = E\left[I_{\mathcal{H}}\left(\mathbf{Y}(\boldsymbol{x})\right) | \boldsymbol{Y}^{m,n} = \boldsymbol{y}^{m,n}\right]. \tag{49}$$

While the authors have found that when $EI_{\mathcal{H}}(\boldsymbol{x})$ can be implemented, it produces Pareto Front approximations that are competitive with those created using $EI_{\mathcal{M}}(\boldsymbol{x})$. *However, the implementation of* $EI_{\mathcal{H}}(\boldsymbol{x})$ *can be difficult for two reasons.* First, it is well-known in the multiobjective function literature that $I_H(\cdot, \cdot)$, and thus $I_{\mathcal{H}}(\mathbf{y}(\boldsymbol{x}))$, requires considerable computational overhead, even for moderately sized m. Therefore, applying the sample average approximation method of Section 4 based on a sample of size S would require S expensive hypervolume calculations. Second, $EI_{\mathcal{H}}(\boldsymbol{x})$ requires the additional specification of the dominated point $\boldsymbol{r}$ to carry out this method. If the objective functions are truly black box functions, $\boldsymbol{r}$ can be difficult to identify. Furthermore, even if one can specify upper bounds for all objectives, the value of $EI_{\mathcal{H}}(\boldsymbol{x})$ will depend on particular choice of the upper bound.

Based on the performance in the examples presented in Section 6 and in other examples that are described in the Supplementary material (see Appendix B), both the expected maximin improvement and the expected hypervolume improvement criteria are highly effective in approximating Pareto Fronts (and Pareto Sets). However, the authors recommend the EMmI/Ind procedure because it is simpler to implement, and requires considerably less computational overhead.

All of $EI_{\mathcal{M}}(\boldsymbol{x})$, $I_{CWPI}(\boldsymbol{x})$, $EI_{\mathcal{P}}(\boldsymbol{x})$ and $EI_{\mathcal{H}}(\boldsymbol{x})$ require scaling each output. In the case of $EI_{\mathcal{M}}(\boldsymbol{x})$, Step 1 of our EMmI Algorithm uses an empirical scaling of each output based on the initial training data; this strategy performed well in all the examples we investigated. However, if one requires a truly scale invariant improvement criterion, the probability of improvement is a possible alternative. Additionally, if one uses the probability of improvement or the centroid-based expected improvement criteria, then the dependence GP model shows some promise.

We conclude by summarizing the several additional research topics identified above that appear to be potentially fruitful, depending on ones' application needs. These include the development of improved prediction models for multiple-output functions, updating strategies that add points in batches rather than one-at-a-time, and the investigation of alternative scale invariant improvement criteria.

Acknowledgments

This research was sponsored, in part, by the National Science Foundation under Agreements DMS-0806134 (The Ohio State University). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would like to thank Professor Angela Dean for helpful discussions.

Appendix A. Proof of Theorem 4.1

To organize the proof, first some known (or easily proved) facts used in the proof are stated. To reiterate notation, suppose $\boldsymbol{A}$ and $\boldsymbol{B}$ are subsets of the same Euclidean space; then $\boldsymbol{A} \prec \boldsymbol{B}$ ($\boldsymbol{A} \preceq \boldsymbol{B}$) provided every $\boldsymbol{a} \in \boldsymbol{A}$ is (weakly) dominated by at least one $\boldsymbol{b} \in \boldsymbol{B}$.

1. $\mathcal{P}_{\mathcal{Y}}^{n} \preceq \mathcal{P}_{\mathcal{Y}}^{n+1}(\mathbf{x})$ since adding any point from the objective space will, at worst, leave the set of nondominated points unchanged. As a consequence,
$$I_{\epsilon+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\mathbf{x})\right) \geq 0,$$
(Zitzler et al., 2003).
2. Suppose that $\boldsymbol{y}(\boldsymbol{x})$ is dominated by at least one point in $\mathcal{P}_{\mathcal{Y}}^{n}$. Then, $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x})) = 0$ and $\mathcal{P}_{\mathcal{Y}}^{n} = \mathcal{P}_{\mathcal{Y}}^{n+1}(\mathbf{x})$.
3. Suppose that $\boldsymbol{y}(\boldsymbol{x})$ is not dominated by any points in $\mathcal{P}_{\mathcal{Y}}^{n}$. Then, $\mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x}) = \{\boldsymbol{y}(\boldsymbol{x})\} \cup G_{\mathcal{Y}}$ where
$$\begin{aligned} G_{\mathcal{Y}} &\equiv \left\{\boldsymbol{y} \in \mathcal{P}_{\mathcal{Y}}^{n} : \boldsymbol{y}(\boldsymbol{x}) \nprec \boldsymbol{y}\right\} \\ &= \left\{\boldsymbol{y} \in \mathcal{P}_{\mathcal{Y}}^{n} : \max_{1\leq k\leq m}\left(y_k(\boldsymbol{x}) - y_k\right) > 0\right\}, \end{aligned} \tag{50}$$
here $G_{\mathcal{Y}}$ can be empty.
4. Suppose that $\boldsymbol{A}$ and $\boldsymbol{B}$ are each subsets of $\mathbb{R}^m$. Then
$$I_{\epsilon+}(\boldsymbol{A}, \boldsymbol{B}) = \max_{\boldsymbol{y}^{(j)}\in\boldsymbol{A}} \min_{\boldsymbol{z}^{(i)}\in\boldsymbol{B}} \max_{1\leq k\leq m}\left(z_k^{(i)} - y_k^{(j)}\right)$$
(Zitzler et al., 2003).
5. From Balling (2003):
   - If fitness $(\boldsymbol{y}(\boldsymbol{x})) > 0$ then $\boldsymbol{y}(\boldsymbol{x})$ is dominated by a vector in $\mathcal{P}_{\mathcal{Y}}^{n}$.
   - If fitness $(\boldsymbol{y}(\boldsymbol{x})) < 0$ then $\boldsymbol{y}(\boldsymbol{x})$ is not dominated by any vector in $\mathcal{P}_{\mathcal{Y}}^{n}$.
   - fitness $(\boldsymbol{y}(\boldsymbol{x})) = 0$ if and only if $\boldsymbol{y}(\boldsymbol{x}) \in \mathcal{P}_{\mathcal{Y}}^{n}$ or $\boldsymbol{y}(\boldsymbol{x})$ is dominated by some element of $\mathcal{P}_{\mathcal{Y}}^{n}$.

Two cases must be considered to prove Theorem 4.1

Case 1 Suppose $\boldsymbol{y}(\boldsymbol{x})$ is dominated by some vector in $\mathcal{P}_{\mathcal{Y}}^{n}$ or that $\boldsymbol{y}(\boldsymbol{x}) \in \mathcal{P}_{\mathcal{Y}}^{n}$.
Then,
$$\mathcal{P}_{\mathcal{Y}}^{n} = \mathcal{P}_{\mathcal{Y}}^{n+1}(\mathbf{x}),$$
so that
$$I_{\epsilon+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\mathbf{x})\right) = I_{\epsilon+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n}\right) = 0.$$
Also, $I_{\mathcal{M}}(\boldsymbol{x}) = 0$, so equality holds.

Case 2 Suppose $\boldsymbol{y}(\boldsymbol{x})$ is not dominated by any vector in $\mathcal{P}_{\mathcal{Y}}^{n}$ and $\boldsymbol{y}(\boldsymbol{x}) \notin \mathcal{P}_{\mathcal{Y}}^{n}$.
Using $\mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x}) = \{\boldsymbol{y}(\boldsymbol{x})\} \cup G_{\mathcal{Y}}$ where $G_{\mathcal{Y}}$ is defined by (50),
$$I_{\epsilon+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\mathbf{x})\right) = \max_{\boldsymbol{y}^{(j)}\in\mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})} \min_{\boldsymbol{z}^{(i)}\in\mathcal{P}_{\mathcal{Y}}^{n}} \max_{1\leq k\leq m}\left(z_k^{(i)} - y_k^{(j)}\right).$$
If $\boldsymbol{y}^{(j)} \in G_{\mathcal{Y}}$, then
$$\begin{aligned} \min_{\boldsymbol{z}^{(i)}\in\mathcal{P}_{\mathcal{Y}}^{n}} \max_{1\leq k\leq m}\left(z_k^{(i)} - y_k^{(j)}\right). &= -\max_{\boldsymbol{z}^{(i)}\in\mathcal{P}_{\mathcal{Y}}^{n}} \min_{1\leq k\leq m}\left(y_k^{(j)} - z_k^{(i)}\right) \\ &= -\text{fitness}(\boldsymbol{y}^{(j)}) \\ &= 0 \end{aligned}$$
because $G_{\mathcal{Y}} \subset \mathcal{P}_{\mathcal{Y}}^{n}$ so that $\boldsymbol{y}^{(j)} \in \mathcal{P}_{\mathcal{Y}}^{n}$. For $\boldsymbol{y}(\boldsymbol{x})$, we have
$$\begin{aligned} \min_{\boldsymbol{z}^{(i)}\in\mathcal{P}_{\mathcal{Y}}^{n}} \max_{1\leq k\leq m}\left(z_k^{(i)} - y_k(\boldsymbol{x})\right). &= -\max_{\boldsymbol{z}^{(i)}\in\mathcal{P}_{\mathcal{Y}}^{n}} \min_{1\leq k\leq m}\left(y_k(\boldsymbol{x}) - z_k^{(i)}\right) \\ &= -\text{fitness}\left(\boldsymbol{y}(\boldsymbol{x})\right) \\ &\geq 0 \end{aligned}$$

as $\boldsymbol{y}(\boldsymbol{x})$ is nondominated by $\mathcal{P}_{\mathcal{Y}}^{n}$. Therefore,

$$\min_{\boldsymbol{z}^{(i)} \in \mathcal{P}_{\mathcal{Y}}^{n}} \max_{1 \le k \le m} \left(z_k^{(i)} - y_k^{(j)} \right)$$

is maximized when $y^{(j)} = \boldsymbol{y}(\boldsymbol{x})$. Therefore, we have

$$\begin{aligned} I_{\epsilon^+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})\right) &= \max_{\boldsymbol{y}^{(j)} \in \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})} \min_{\boldsymbol{z}^{(i)} \in \mathcal{P}_{\mathcal{Y}}^{n}} \max_{1 \le k \le m} \left(z_k^{(i)} - y_k^{(j)} \right) \\ &= -\max_{\boldsymbol{z}^{(i)} \in \mathcal{P}_{\mathcal{Y}}^{n}} \min_{1 \le k \le m} \left(y_k(\boldsymbol{x}) - z_k^{(i)} \right) \\ &= -\max_{\boldsymbol{z}^{(i)} \in \mathcal{P}_{\mathcal{Y}}^{n}} \min_{1 \le k \le m} \left(y_k(\boldsymbol{x}) - z_k^{(i)} \right) \times 1_{\left[-\max_{\boldsymbol{z}^{(i)} \in \mathcal{P}_{\mathcal{Y}}^{n}} \min_{1 \le k \le m} \left(y_k(\boldsymbol{x}) - z_k^{(i)} \right) > 0 \right]} \\ &= I_{\mathcal{M}}\left(\boldsymbol{y}(\boldsymbol{x})\right). \end{aligned}$$

Note that the indicator function in the formula for $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$ must equal 1 because $-\text{fitness}(\boldsymbol{y}(\boldsymbol{x})) > 0$, since $\boldsymbol{y}(\boldsymbol{x})$ is nondominated and not an element of $\mathcal{P}_{\mathcal{Y}}^{n}$. □

Appendix B. Supplementary material

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.csda.2015.08.011.

References

Audet, C., Dennis Jr., J.E., 2006. Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. 17, 188–217.

Audet, C., Savard, G., Zghal, W., 2010. A mesh adaptive direct search algorithm for multiobjective optimization. European J. Oper. Res. 204, 545–556.

Balling, R., 2003. The maximin fitness function: Multiobjective city and regional planning. In: Fonseca, C., Fleming, P., Zitzler, E., Deb, K., Thiele, L. (Eds.), Evolutionary Multi-Criterion Optimization. Springer, pp. 1–15.

Banerjee, S., Gelfand, A.E., Finley, A.O., Sang, H., 2008. Gaussian predictive process models for large spatial data sets. J. R. Stat. Soc. Ser. B 70, 825–848.

Bautista, D.C., 2009. A Sequential Design for Approximating the Pareto Front Using the Expected Pareto Improvement Function (Ph.D. thesis). Department of Statistics, The Ohio State University, Columbus, Ohio USA.

Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A., 2006. Evolutionary Algorithms for Solving Multi-Objective Problems. In: Genetic and Evolutionary Computation, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Deb, K., Thiele, L., Laumanns, M., Zitzler, E., 2005. Scalable test problems for evolutionary multi-objective optimization. In: Abraham, A., Jain, R., Goldberg, R. (Eds.), Evolutionary Multiobjective Optimization: Theoretical Advances and Applications. Springer, pp. 105–145. chapter 6.

Emmerich, M.T., Giannakoglou, K.C., Naujoks, B., 2006. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. IEEE Trans. Evol. Comput. 10.

Fonseca, C., Fleming, P., 1995. Multiobjective genetic algorithms made easy: selection sharing and mating restriction. In: Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995. GALESIA. First International Conference on (Conf. Publ. No. 414), pp. 45–52.

Forrester, A., Sobester, A., Keane, A., 2008. Engineering Design via Surrogate Modelling: A Practical Guide. Wiley, Chichester, UK.

Fricker, T.E., Oakley, J.E., Urban, N.M., 2013. Multivariate gaussian process emulators with nonseparable covariance structures. Technometrics 55, 47–56. http://dx.doi.org/10.1080/00401706.2012.715835.

Gelfand, A.E., Schmidt, A.M., Banerjee, S., Sirmans, C., 2004. Nonstationary multivariate process modeling through spatially varying coregionalization. Test (Madrid) 13, 263–294.

Huang, D., Allen, T., Notz, W., Zeng, N., 2006. Global optimization of stochastic black-box systems via sequential kriging meta-models. J. Global Optim. 34, 441–466. http://dx.doi.org/10.1007/s10898-005-2454-3.

Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black–box functions. J. Global Optim. 13, 455–492.

Keane, A.J., 2006. Statistical improvement criteria for use in multiobjective design optimization. AIAA J. 44, 879–891.

Knowles, J., 2006. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjetive optimization problems. IEEE Trans. Evol. Comput. 10, 50–66.

Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P., 1989. Design and analysis of computer experiments. Statist. Sci. 4, 409–423.

Santner, T.J., Williams, B.J., Notz, W.I., 2003. The Design and Analysis of Computer Experiments. Springer Verlag, New York.

Schonlau, M., 1997. Computer Experiments and Global Optimization (Ph.D. thesis). Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, CA.

Shapiro, A., 2003. Monte Carlo sampling methods. In: Ruszczynski, A., Shapiro, A. (Eds.), Stochastic Programming. In: Handbooks in Operations Research and Management Science, vol. 10. Elsevier, pp. 353–425.

Ver Hoef, J.M., Cressie, N., 1993. Multivariate spatial prediction. Math. Geol. 25, 219–240.

Williams, B.J., Santner, T.J., Notz, W.I., 2000. Sequential design of computer experiments to minimize integrated response functions. Statist. Sinica 10, 1133–1152.

Williams, B.J., Santner, T.J., Notz, W.I., Lehman, J.S., 2010. Sequential design of computer experiments for constrained optimization. In: Kneib, T., Tutz, G. (Eds.), Statistical Modelling and Regression Structures: Festschrift in Honour of Ludwig Fahrmeir. Springer Verlag: Berlin Heidelberg, Berlin and Heidelberg, pp. 449–472.

Zitzler, E., Knowles, J., Thiele, L., 2008. Quality assessment of Pareto set approximations. In: Branke, J., Deb, K., Miettinen, K., Slowinski, R. (Eds.), Multiobjective Optimization: Interactive and Evolutionary Approaches. Springer, pp. 373–404.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V., 2003. Performance assessment of multiobjective optimizers: An analysis and review. IEEE Trans. Evol. Comput. 7, 117–132.