Optimization and Engineering manuscript No.
(will be inserted by the editor)

Multiobjective Optimization of Expensive Black-Box Functions via Expected Maximin Improvement

Joshua D. Svenson · Thomas J. Santner

Received: date / Accepted: date

Joshua D. Svenson
The Ohio State University Department of Statistics, 1958 Neil Avenue, 404 Cockins Hall, Columbus, OH 34210-1247 E-mail: svenson.4@buckeyemail.osu.edu

Thomas J. Santner
The Ohio State University Department of Statistics, 1958 Neil Avenue, 404 Cockins Hall, Columbus, OH 34210-1247 E-mail: santner.1@osu.edu

Abstract Many engineering design optimization problems contain multiple objective functions all of which it is desired to minimize, say. One approach to solving this problem is to identify those inputs to the objective functions that produce an output (vector) on the Pareto Front; the inputs that produce outputs on the Pareto Front form the Pareto Set. This paper proposes a method for identifying the Pareto Front and the Pareto Set when the objective functions are *expensive to compute*. The method replaces the objective function evaluations by a rapidly computable approximator based on an interpolating Gaussian process (GP) model. It sequentially selects new input sites guided by an improvement function; the next input to evaluate each output is that vector which maximizes the conditional expected value of this improvement function given the current data. The method introduced in this paper provides two advances within this framework. First, it proposes an improvement function based on the modified maximin fitness function. Second, it uses a family of GP models that allow for dependent output functions but which permits zero covariance should the data be consistent with a model of no association. GP models with dependent component functions have the potential to provide more precise predictions of competing objectives than independent GPs. A closed-form expression is derived for the conditional expectation of the proposed improvement function when there are two objective functions; simulation is used to evaluate this expectation 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.

Keywords Computer experiment · Kriging · Gaussian Process · Pareto optimization · Nonseparable model · Computer simulator

1 Introduction

This paper proposes an algorithm for sequentially designing a sequence of inputs at which to evaluate a set of functions so to identify the Pareto Front of this set of functions and the Pareto Set of inputs producing function values on the Pareto front. Stated in detail, it assumed that there are m functions of interest which are denoted by $y_1(\boldsymbol{x})$,...,$y_m(\boldsymbol{x})$). The number of inputs to the functions is denoted by d and $\boldsymbol{x} = (x_1, \ldots, x_d)$ denotes a generic input of the function. The input space for $\boldsymbol{x}$ is denoted by $\mathcal{X}$ which, therefore, subset of $\mathrm{I\!R}^d$. The objective is to find the complement of the set of *dominated* inputs $\boldsymbol{x}$. The input $\boldsymbol{x}_1 \in \mathcal{X}$ is said to be *dominated* if there is an input $\boldsymbol{x}_2 \in \mathcal{X}$ with $\boldsymbol{x}_2 \neq \boldsymbol{x}_1$ for which if $y_i(\boldsymbol{x}_1) \leq y_i(\boldsymbol{x}_2)$ for all $i = 1, \ldots, m$. Geometrically, $\boldsymbol{y}(\boldsymbol{x}_1)$ dominates $\boldsymbol{y}(\boldsymbol{x}_2)$ if $\boldsymbol{y}(\boldsymbol{x}_1)$ is to the "southwest" of $\boldsymbol{y}(\boldsymbol{x}_2)$. Stated in other words, the goal is to find the set of all *nondominated* inputs $\boldsymbol{x} \in \mathcal{X}$; the latter set is called the *Pareto Set.* The set of $y_1(\boldsymbol{x})$,...,$y_m(\boldsymbol{x})$) corresponding to inputs $\boldsymbol{x}$ in the Pareto set is termed the *Pareto Front.* Stating the goal geometrically, we propose an algorithm which adaptively determines a sequence of inputs $\boldsymbol{x}$ that identifies the "southwest" boundary of the set of function values $\boldsymbol{y}(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{X}$.

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 the Pareto front and the Pareto set, such as the weighted sum method, the $\epsilon-$constrained method, and multiobjective evolutionary algorithms, require a large number of function evaluations. This paper proposes methodology for cases when $\boldsymbol{y}(\cdot)$ is expensive-to-compute and thus a budget with a severely *limited number of evaluations are possible.*

Broadly, the approach used in this paper is to build a cheap-to-compute surrogate, a "meta-model," for $\boldsymbol{y}(\cdot)$, and then use the surrogate to guide the search for nondominated points. The authors will employ an interpolator for $\boldsymbol{y}(\cdot)$ based on a *Gaussian process* (GP) model (see [15]). While the use of interpolator/expected improvement has been considered previously in the literature (see [16], [11], [12], [6], and [13]) the methodology proposed in this paper provides two key improvements in its detailed implementation over previous research. First, the improvement criterion is based on the maximin fitness function (see [2]). Second, it is the only proposed multiobjective expected improvement approach that considers the use of stochastic prediction models which allow for dependence *among* the components of $\boldsymbol{y}(\cdot)$; depending on the application and dependence model, there have applications where allowing dependence can lead to improved procedure performance ([**?**], [**?**], []).

The remainder of this paper is organized as follows. To provide context, Section 2 reviews the expected improvement approach proposed in [16] and [11] 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 [12]. Finally, Section 7 contains recommendations as to which methods should be used in practice, compares the proposed approach to the hypervolume-based method of [6], 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, the present section introduces the key ideas for the simpler problem of minimizing a single (expensive) *real-valued* function defined on a d-dimensional input space $\mathcal{X}$. The method described is due to [16] and [11] who introduced a method for minimizing $y(\cdot)$ based on a Gaussian stochastic process model which they called the “efficient global optimization” (EGO) algorithm. Given the probabilistic assessment of $y(\boldsymbol{x})$ that is provided by the GP model, the authors compute the (conditional) expectation of a heuristically selected improvement function given the current data in order to determine the information in each potential $\boldsymbol{x}$ about the global minimum of $y(\cdot)$.

Suppose that $y(\boldsymbol{\cdot})$ has been evaluated at each input in $\mathcal{D}_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathcal{X}$. 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

$$R\left(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\rho}, \boldsymbol{\theta}\right) = \exp\left\{\sum_{i=1}^{d} \theta_i \left|x_i - x_i'\right|^{\rho_i}\right\}. \tag{1}$$

The parameters $(\beta, \sigma^2, \theta_1, \ldots, \theta_d, \rho_1, \ldots, \rho_d)$ are unknown and must be estimated to complete specification of the GP model. This GP provides the basis for interpolation of $y(\cdot)$ and uncertainty assessment of the predicted values.

It can be shown that the best linear unbiased predictor (BLUP) of $y(\boldsymbol{x})$ is the mean of $Y(\boldsymbol{x})$ conditional on $Y^n = (Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n))$ equal to $\boldsymbol{y}^n$ ([14]). When the covariance parameters σ^2, $\boldsymbol{\theta}$, $\boldsymbol{\rho}$ are known, the BLUP of $y(\boldsymbol{x})$ is

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

where $\boldsymbol{R} = (R_{ij})$ is the $n \times n$ matrix with $R_{ij} = R\left(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\rho}, \boldsymbol{\theta}\right)$, $\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{x}) = (r_i)$ is the $n \times 1$ vector with $r_i = R\left(\boldsymbol{x}, \boldsymbol{x}_i; \boldsymbol{\rho}, \boldsymbol{\theta}\right)$, and

$$\widehat{\beta} = \frac{\mathbf{1}^T\boldsymbol{R}^{-1}\boldsymbol{y}^n}{\mathbf{1}^T\boldsymbol{R}^{-1}\mathbf{1}}. \tag{3}$$

The mean square prediction error (MSPE) of $\widehat{y}(\boldsymbol{x})$, which is used to assess its accuracy is defined to be $s^2(\boldsymbol{x}) \equiv E\{(Y(\boldsymbol{x}) - \widehat{y}(\boldsymbol{x}))^2|\boldsymbol{y}^n\}$ which can be shown to be

$$s^2(\boldsymbol{x}) = \sigma^2 \left[1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r} + \frac{\left(1 - \mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}\right)^2}{\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}} \right]. \tag{4}$$

In practice, σ^2, $\boldsymbol{\theta}$ and $\boldsymbol{\rho}$ are unknown. The frequentist approach to addressing this situation is to estimate the unknown parameters by, say, maximum likelihood and apply formulas (2) and (4) with estimated parameters producing an "empirical" BLUP and MSPE. Then, conditional on $\boldsymbol{y}^n$, the process $Y(\boldsymbol{x})$ is (approximately) a normally distributed random variable with mean $\widehat{y}(\boldsymbol{x})$ and variance $s^2(\boldsymbol{x})$. An alternative modeling approach to the problem of unknown parameters is the Bayesian solution in which priors are identified that embody knowledge about these quantities. While both approaches can be applied below, this paper will follow [16] and [11] and substitute estimated parameters into the BLUP and MSPE formulas.

The EGO algorithm is based on a heuristically selected improvement function defined for each new potential input $\boldsymbol{x}$. [16] and [11] selected the theoretical improvement function

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

where y^n_{min} is the smallest element in $\boldsymbol{y}^n$ and 1_E is 1 if E is true and 1_E is 0 if E is false. 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 expection of $I(Y(\boldsymbol{x}))$ given $\boldsymbol{y}^n$, i.e., $EI(\boldsymbol{x}) = E\left\{I(Y(\boldsymbol{x}))|\boldsymbol{y}^n\right\}$, is approximately

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

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 (which 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, $\boldsymbol{\theta}$ and $\boldsymbol{\rho}$ based on $\boldsymbol{y}^n$.
3. Find $\boldsymbol{x}^{n+1} \in \arg\max EI(\boldsymbol{x})$.
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

Let $\boldsymbol{y}(\boldsymbol{x}) = (y_1(\boldsymbol{x}), \ldots, y_m(\boldsymbol{x}))$ denote an m-dimensional black box function with a d-dimensional input $\boldsymbol{x}$ in $\mathcal{X}$. Throughout it will be assumed that $\boldsymbol{y}(\boldsymbol{x})$ can be described as a draw from an m-variate Gaussian process $\boldsymbol{Y}(\boldsymbol{x})$. This paper considers $\boldsymbol{Y}(\boldsymbol{x})$ processes of the form

$$\boldsymbol{Y}(\boldsymbol{x}) = \boldsymbol{\beta} + \boldsymbol{A}\boldsymbol{Z}(\boldsymbol{x}) \tag{7}$$

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

$$\boldsymbol{\beta} = (\beta_1 \ldots \beta_m)^T , \tag{8}$$

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 with *zero mean* and *unit variance*. The mean of the process $\boldsymbol{Y}(\cdot)$ is $\boldsymbol{\beta}$; its variance-covariance is determined by $\boldsymbol{A}$.

In more detail, it is assumed that the process $Z_i(\boldsymbol{x})$ is assumed to have correlation function of the form

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

Under these assumptions it is straightforward to show

$$Cov(\boldsymbol{Y}(\boldsymbol{x}), \boldsymbol{Y}(\boldsymbol{x}')) = \boldsymbol{A} diag\left(R(\boldsymbol{x} - \boldsymbol{x}'; \boldsymbol{\theta}_1), \ldots, R(\boldsymbol{x} - \boldsymbol{x}'; \boldsymbol{\theta}_m)\right) \boldsymbol{A}^T, \tag{10}$$

so that, when $\boldsymbol{x} = \boldsymbol{x}'$,

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

Thus the model states that the components of $\boldsymbol{Y}(\boldsymbol{x})$ have a correlation and variance structure that is the *same* for all $\boldsymbol{x}$ and that the component process $Y_i(\boldsymbol{x})$ is stationary with variance $\sum_{j=1}^{m} a_{ij}^2$.

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 $\boldsymbol{y}^{m,n} = (\boldsymbol{y}^T(\boldsymbol{x}_1), \ldots, \boldsymbol{y}^T(\boldsymbol{x}_n))^T$ denote the associated $mn \times 1$ stacked vector of outputs and $\boldsymbol{Y}^{m,n}$ the associated process values. Let Σ_{mn} denote the $mn \times mn$ covariance matrix of $\boldsymbol{Y}^{m,n}$; it is easy to compute that Σ_{mn} is

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

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

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

This paper will consider two general choices of form for $\boldsymbol{A}$. The first form assumes that $\boldsymbol{A}$ is a diagonal matrix with positive entries. This assumption is equivalent to fitting a separate Gaussian process to each $y_i(\cdot)$, $1 \leq i \leq m$ and hence is termed the *independence* model. The second assumed form for $\boldsymbol{A}$ is any symmetric, positive-definite matrix. This assumption permits dependence *among* the various outputs and hence is termed the *nonseparable dependence* model. The nonseparable dependence model, $\boldsymbol{A}$ is to be thought of as the unique matrix square root of $\varSigma_0$ (which can be calculated via eigen decomposition). Note that while some geostatistics literature proposes treating $\boldsymbol{A}$ as the lower triangular Cholesky decomposition of $\varSigma_0$ (see [10] and [3]), [9] shows that such a specification induces artificial asymmetry into the covariance structure and therefore argues that the eigen decomposition is more appropriate for modeling functions having no *a priori* hierarchy of dependence.

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

$$\left[\left(\begin{matrix} \boldsymbol{Y}(\boldsymbol{x}_0) \\ \boldsymbol{Y}^{m,n} \end{matrix} \right) \middle| \boldsymbol{\beta} \right] \sim N \left(\left[\begin{matrix} \boldsymbol{I}_m \\ \mathcal{F} \end{matrix} \right] \boldsymbol{\beta}, \left[\begin{matrix} \varSigma_0 & \varSigma_{0,m,n} \\ \varSigma_{0,m,n}^T & \varSigma_{m,n} \end{matrix} \right] \right), \tag{14}$$

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

$$\begin{aligned} &[\boldsymbol{Y}(\boldsymbol{x}_0) | \boldsymbol{Y}^{m,n} = \boldsymbol{y}^{m,n}, \boldsymbol{\beta}] \\ &\sim N \left(\boldsymbol{\beta} + \varSigma_{0,m,n} \varSigma_{m,n} (\boldsymbol{y}^{m,n} - \mathcal{F}\boldsymbol{\beta}), \varSigma_0 - \varSigma_{0,m,n} \varSigma_{m,n}^{-1} \varSigma_{0,m,n}^T \right). \end{aligned} \tag{15}$$

Hence integrating out $\boldsymbol{\beta}$ with respect to the standard non-informative uniform prior yields

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

where

$$\widehat{\boldsymbol{y}}(\boldsymbol{x}_0) = \widehat{\boldsymbol{\beta}}_{\text{GLS}} + \varSigma_{0,m,n} \varSigma_{m,n} (\boldsymbol{y}^{m,n} - \mathcal{F} \widehat{\boldsymbol{\beta}}_{\text{GLS}}), \text{ with} \tag{17}$$

$$\widehat{\boldsymbol{\beta}}_{\text{GLS}} = (\mathcal{F}^T \varSigma_{m,n}^{-1} \mathcal{F})^{-1} \mathcal{F}^T \varSigma_{m,n}^{-1} \boldsymbol{y}^{m,n}, \tag{18}$$

and estimated prediction uncertainty

$$\begin{aligned} \boldsymbol{S}(\boldsymbol{x}_0) = \varSigma_0 &- \varSigma_{0,m,n} \varSigma_{m,n}^{-1} \varSigma_{0,m,n}^T + (\boldsymbol{I}_m - \varSigma_{0,m,n} \varSigma_{m,n}^{-1} \mathcal{F}) \\ &\times (\mathcal{F}^T \varSigma_{m,n}^{-1} \mathcal{F})^{-1} \times (\boldsymbol{I}_m - \varSigma_{0,m,n} \varSigma_{m,n}^{-1} \mathcal{F})^T \ . \end{aligned} \tag{19}$$

When $\boldsymbol{A}$ (assumed to be one of the two model forms) and $\boldsymbol{\theta}$ are unknown, this paper estimates them using restricted maximum likelihood (REML) and their estimates are plugged into (17), (18), and (19). Specifically, the estimated $\boldsymbol{A}$ and $\boldsymbol{\theta}$ are

$$\begin{aligned} \left(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{\theta}} \right) \in \text{argmax} &\left\{ -\frac{1}{2} \log \left(|\varSigma_{m,n}| \right) \frac{1}{2} \log \left(\mathcal{F}^T \varSigma_{m,n}^{-1} \mathcal{F} \right) \right. \\ &\left. - \frac{1}{2} \left(\boldsymbol{y}^{m,n} - \mathcal{F} \widehat{\boldsymbol{\beta}}_{\text{GLS}} \right)^T \varSigma_{m,n}^{-1} \left(\boldsymbol{y}^{m,n} - \mathcal{F} \widehat{\boldsymbol{\beta}}_{\text{GLS}} \right) \right\} \end{aligned} \tag{20}$$

where the maximum is over the assumed form for $\boldsymbol{A}$.

4 The Expected Maximin Improvement Function

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

The proposed generalization of $I(y(\boldsymbol{x}))$ is the *maximin improvement function*

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

where the indicator function 1_E is 1 or 0 according as the event E is true or not.

A non-truncated version of (21) (i.e., lacking the indicator function component) was introduced in [4]. Both improvement functions are based on the "modified maximin fitness function"

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

presented in [2] and was originally introduced as a component of a multiobjective evolutionary algorithm. The expected improvement function

$$EI_{\mathcal{M}}(\boldsymbol{x}) = E\{I_{\mathcal{M}}\left(\boldsymbol{Y}(\mathbf{x})\right) | \boldsymbol{Y}^{n,m} = \boldsymbol{y}^{n,m}\} \tag{23}$$

is used below.

Before describing the calculation of $EI_{\mathcal{M}}(\boldsymbol{x})$, some properties of $I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x})\right)$ will be discussed. First, when $m = 1$, it is straightforward to show that $I_{\mathcal{M}}(y(\boldsymbol{x})) = I(y(\boldsymbol{x}))$ so that single-objective improvement function is just a special case of the maximin improvement function. As noted by [2], it is easy to show that $I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x})\right) > 0$ if and only if $\boldsymbol{y}(\mathbf{x})$ is not dominated by any vector in $\mathcal{P}^n_{\mathcal{Y}}$, and $I_{\mathcal{M}}\left(\mathbf{x}\right) = 0$ if and only if $\boldsymbol{y}(\mathbf{x})$ is dominated by a vector in $\mathcal{P}^n_{\mathcal{Y}}$. Additionally, $I_{\mathcal{M}}\left(\mathbf{x}\right)$ is monotonic with respect to Pareto dominance, in the sense that $I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x})\right) \geq I_{\mathcal{M}}\left(\boldsymbol{y}(\mathbf{x}')\right)$ provided that $\boldsymbol{y}(\mathbf{x}) \succeq \boldsymbol{y}(\mathbf{x}')$.

Lastly, it is noted that $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$ is closely related to the additive Binary-ϵ indicator; the Binary-ϵ indicator is a popular Pareto set approximation quality indicator introduced in [19]. This indicator allows one to compare two Pareto Front approximations B and C in the objective space. Roughly, the additive binary-ϵ indicator of C relative to B measures how much "better" C is than B in terms of dominance; specifically, the additive binary-ϵ indicator of C relative to B is the smallest real number that must be added to all vectors in C (thus "worsening" C) so that B dominates the degraded C, i.e.,

$$I_{\epsilon^+}\left(B, C\right) = \inf_{\epsilon \in \mathbb{R}} \left\{\forall\, \boldsymbol{y}^2 \in C\, \exists\, \boldsymbol{y}^1 \in B : y^1_i \leq \epsilon + y^2_i \;\; \forall \;\; i = 1, \ldots, m\right\} . \tag{24}$$

To describe the relationship between the expected maximin improvement and the additive binary-ϵ indicator, first let $\mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})$ be the current Pareto front if $\boldsymbol{y}^{m,n}$ is augmented by $\boldsymbol{y}(\boldsymbol{x})$. Then, one could think of $I_{\epsilon^+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})\right)$ as quantifying how much better $\mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})$ is than $\mathcal{P}_{\mathcal{Y}}^{n}$, i.e., how much $\boldsymbol{y}(\boldsymbol{x})$ improves upon our current best Pareto front approximation. 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, replace $\boldsymbol{y}(\boldsymbol{x})$ by $\boldsymbol{Y}(\boldsymbol{x})$ in $I_{\epsilon^+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})\right)$ and choose $\boldsymbol{x}$ to maximize

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

However, as the following theorem shows, such a strategy is actually equivalent to the expected maximin fitness approach.

Theorem 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, $I_{\epsilon^+}\left(\mathcal{P}_{\mathcal{Y}}^{n}, \mathcal{P}_{\mathcal{Y}}^{n+1}(\boldsymbol{x})\right) = I_{\mathcal{M}}\left(\boldsymbol{y}(\boldsymbol{x})\right)$.*

Proof See Online Reseource 1

Therefore, using the maximin improvement function to control the search for the Pareto front is essentially equivalent to using the additive binary-ϵ indicator to control the search for the Pareto front.

There is a final practical issue one must resolve before calculating $EI_{\mathcal{M}}(\boldsymbol{x})$. The maximin improvement function and its expectation both depend upon the scaling of the various outputs $y_1(\cdot), \ldots, y_m(\cdot)$. Therefore, the outputs are empirically scaled so that, for each objective $y_i(\cdot)$, min $\{y_i(\boldsymbol{x}_1), \ldots, y_i(\boldsymbol{x}_n)\} = 0$ and max $\{y_i(\boldsymbol{x}_1), \ldots, y_i(\boldsymbol{x}_n)\} = 1$.

4.1 Calculation of $EI_{\mathcal{M}}(\boldsymbol{x})$

When $m = 2$ there is a nearly a closed-form expression for $EI_{\mathcal{M}}(\boldsymbol{x})$ that can be implemented quite accurately in computer code. When $m \geq 3$, one must use Monte Carlo methods to estimate $EI_{\mathcal{M}}(\boldsymbol{x})$ (detailed in next section).

Returning to the $m = 2$ case where $\boldsymbol{Y}(\boldsymbol{x}) = (Y_1(\boldsymbol{x}), Y_2(\boldsymbol{x}))$ has conditional mean and covariance

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

and

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

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

Without loss of generality, assume that the points are labeled so that $y_1(\boldsymbol{x}_1^*) \leq \ldots \leq y_1(\boldsymbol{x}_p^*)$. As a consequence of the fact that $\mathcal{P}_{\mathcal{Y}}^{n}$ cannot contain any dominated points, it must be the case that $y_2(\boldsymbol{x}_1^*) \geq \ldots \geq y_2(\boldsymbol{x}_p^*)$. 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 $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$

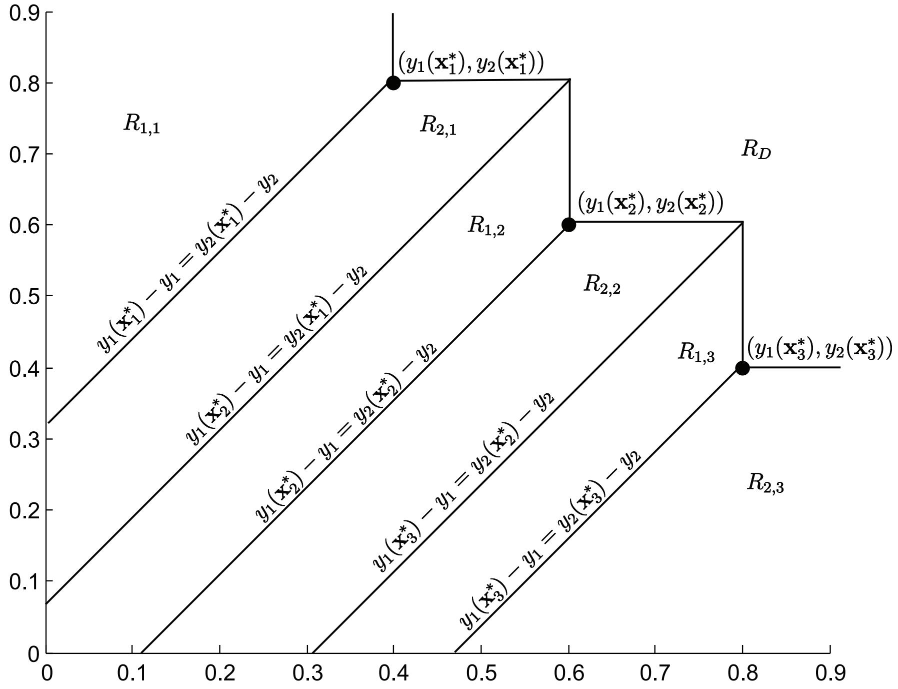
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
$(y_1(\mathbf{x}_1^*), y_2(\mathbf{x}_1^*))$
$(y_1(\mathbf{x}_2^*), y_2(\mathbf{x}_2^*))$
$(y_1(\mathbf{x}_3^*), y_2(\mathbf{x}_3^*))$
$R_{1,1}$
$R_{2,1}$
R_D
$R_{1,2}$
$R_{2,2}$
$R_{1,3}$
$R_{2,3}$
$y_1(\mathbf{x}_1^*) - y_1 = y_2(\mathbf{x}_1^*) - y_2$
$y_1(\mathbf{x}_2^*) - y_1 = y_2(\mathbf{x}_1^*) - y_2$
$y_1(\mathbf{x}_2^*) - y_1 = y_2(\mathbf{x}_2^*) - y_2$
$y_1(\mathbf{x}_3^*) - y_1 = y_2(\mathbf{x}_2^*) - y_2$
$y_1(\mathbf{x}_3^*) - y_1 = y_2(\mathbf{x}_3^*) - y_2$

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

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

$$\begin{aligned}&\Big\{ \big(y_i, y_{k(i)}\big) \;:\; y_i \le y_i(\boldsymbol{x}_j^*),\\ &y_{k(i)}(\boldsymbol{x}_j^*) - y_i(\boldsymbol{x}_j^*) + y_i \le y_{k(i)} \le y_{k(i)}(\boldsymbol{x}_{h(i,j)}^*) - y_i(\boldsymbol{x}_j^*) + y_i \Big\}\end{aligned} \tag{26}$$

and

$$R_D = \big\{ (y_1, y_2) \;:\; \{(y_1, y_2)\} \prec \mathcal{P}_{\mathcal{Y}}^n \big\}\,. \tag{27}$$

Figure 1 shows an example of this set of regions.

In the general case, $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$ is equal to $y_i(\boldsymbol{x}_j^*) - y_i$ for $\boldsymbol{x} \in R_{i,j}$, while $I_{\mathcal{M}}(\boldsymbol{y}(\boldsymbol{x}))$ is equal to 0 for $\boldsymbol{x} \in R_D$. Therefore, letting

$$Int_{i,j} = \int_{-\infty}^{y_i(\boldsymbol{x}_j^*)} \int_{y_{k(i)}(\boldsymbol{x}_j^*) - y_i(\boldsymbol{x}_j^*) + y_i}^{y_{k(i)}(\boldsymbol{x}_{h(i,j)}^*) - y_i(\boldsymbol{x}_j^*) + y_i} \big[y_i(\boldsymbol{x}_j^*) - y_i\big]\, f(y_1, y_2) dy_{k(i)} dy_i \tag{28}$$

where $i = 1, 2$, $j = 1, \ldots, p$, and $f(y_1, y_2)$ is the bivariate conditional normal probability density function of $\boldsymbol{Y}(\boldsymbol{x})$ gives

$$EI_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{2} \sum_{j}^{p} Int_{i,j}. \tag{29}$$

Finally, accounting for the different upper and lower bounds for each $Int_{i,j}$, we can prove the following theorem:

Theorem 2

$$EI_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i=1}^{2}\sum_{j=1}^{p}\big(Int^1_{i,j}(\boldsymbol{x}) + Int^2_{i,j}(\boldsymbol{x}) + Int^3_{i,j}(\boldsymbol{x})\big) \tag{30}$$

where

$$\begin{aligned}
Int^1_{i,j}(\boldsymbol{x}) &= s_i(\boldsymbol{x})\phi\left(\frac{d(i,j)}{s_i(\boldsymbol{x})}\right)\\
&\times\left[\Phi\left(\frac{-d(k(i),j)+\rho(\boldsymbol{x})s_{k(i)}(\boldsymbol{x})d(i,j)/s_i(\boldsymbol{x})}{\sqrt{(1-\rho^2(\boldsymbol{x}))s^2_{k(i)}(\boldsymbol{x})}}\right)\right.\\
&\left.-\,\Phi\left(\frac{-d(k(i),h(i,j))+\rho(\boldsymbol{x})s_{k(i)}(\boldsymbol{x})d(i,j)/s_i(\boldsymbol{x})}{\sqrt{(1-\rho^2(\boldsymbol{x}))s^2_{k(i)}(\boldsymbol{x})}}\right)\right],
\end{aligned}$$

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

and

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

with constants

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

Proof We prove this result in Appendix **??** by showing that $Int_{i,j} = Int_{i,j}^1 + Int_{i,j}^2 + Int_{i,j}^3$.

5 An Algorithm for Approximating the Pareto Front and Set

First, an outline of our proposed multiobjective optimization algorithm based on the expected maximin fitness function will be stated. Then some of the computational details and issues in its implementation 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 $\boldsymbol{y}^{m,n} = (\boldsymbol{y}^T(\boldsymbol{x}_1), \ldots, \boldsymbol{y}^T(\boldsymbol{x}_n))^T$. Empirically scale the ouputs 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 $\boldsymbol{y}^{m,n}$ (or another method such as maximum likelihood).
3. Calculate the current Pareto set $\mathcal{P}_{\mathcal{X}}^n$ and current Pareto front $\mathcal{P}_{\mathcal{Y}}^n$. These are the nondominated inputs and outputs, respectively, in $\mathcal{D}_n$ and $\boldsymbol{y}^{m,n}$.
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
$$\boldsymbol{y}^{m,n+1} = (\boldsymbol{y}^T(\boldsymbol{x}_1), \ldots, \boldsymbol{y}^T(\boldsymbol{x}_n), \boldsymbol{y}^T(\boldsymbol{x}_{n+1}))^T \tag{31}$$
until the computational budget has been exhausted or other stopping criteria met.

While the computational budget will often be the reason that determines stopping, there are other possible stopping criteria. Perhaps the most important of these is to use the maximum expected improvement, max $EI_{\mathcal{M}}(\boldsymbol{x})$ given the data at collected to date. If the maximum expected improvement is sufficiently small, then one might consider terminating sampling ([16]). 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 ([**?**]).

In the examples below, the initial space-filling design was taken to be a maximin LHD constructed using the MATLAB function `bestlh`, available in on-line supplementary material for [8]. In the case of independent outputs, REML estimates of the covariance parameters (and then $\widehat{\boldsymbol{y}}(\boldsymbol{x})$, and $\boldsymbol{S}(\boldsymbol{x})$) were obtained using MPErK software, which can be obtained by contacting the second author. In the nonseparable dependence model, the MATLAB function `ga`, also available as a component of the on-line supplementary material to [8], was used to obtain the initial estimate of $\boldsymbol{\theta}$ and $\boldsymbol{A}$; these values were taken to be the initial point in an application of the MATLAB `fmincon` function to produce the final estimates of $\boldsymbol{\theta}$ and $\boldsymbol{A}$. These estimates are then used to calculate $\boldsymbol{y}(\boldsymbol{x})$, and $\boldsymbol{S}(\boldsymbol{x})$. 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 $\mathcal{P}^n_{\mathcal{X}}$ and $\mathcal{P}^n_{\mathcal{Y}}$.

NOMADm, Mark Abramson's MATLAB implementation of a mesh adaptive direct search (MADS) algorithm (see [1]) available at `http://www.gerad.ca/NOMAD/Abramson/nomadm.html`, was used to optimize $EI_{\mathcal{M}}\left(\boldsymbol{x}\right)$. When $m = 2$, $EI_{\mathcal{M}}\left(\boldsymbol{x}\right)$ can be directly calculated and was optimized in a straightforward manner. When $m \geq 3$, $EI_{\mathcal{M}}\left(\boldsymbol{x}\right)$ was optimized via sample average approximation (SAA, described in [17]). The idea of SSA is to construct an approximation to $EI_{\mathcal{M}}\left(\boldsymbol{x}\right)$ 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 $N(\boldsymbol{0}_m, \mathrm{I}_m)$ 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 distribtion of $\boldsymbol{Y}(\boldsymbol{x})$ given the data. The *sample average function* $\widehat{EI_{\mathcal{M}}}\left(\boldsymbol{x}\right) = \frac{1}{S}\sum_{s=1}^{S} I^s_{\mathcal{M}}(\boldsymbol{x})$ is a *deterministic* function for a particular realization of the random sample $\boldsymbol{Z}^1, \ldots, \boldsymbol{Z}^S$) where

$$
\begin{aligned}
I^s_{\mathcal{M}}\left(\mathbf{x}\right) = & -\max_{\mathbf{x}_i \in \mathcal{P}^n_{\mathcal{X}}} \min_{j=1,\ldots,m} \left(Y^s_j(\mathbf{x}) - y_j(\mathbf{x}_i)\right) \\
& \times 1\left[-\max_{\mathbf{x}_i \in \mathcal{P}^n_{\mathcal{X}}} \min_{j=1,\ldots,m} \left(Y^s_j(\mathbf{x}) - y_j(\mathbf{x}_i)\right) < 0\right].
\end{aligned}
\tag{32}
$$

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

6 Examples

The performance of the expected maximin improvement (EMMI) will be compared with that of two other competing improvement criteria.The first criterion from [12] proposed choosing $\boldsymbol{x}^{n+1}$ to maximize the conditional probability that $\boldsymbol{Y}(\boldsymbol{x})$ is not dominated by the Pareto front, given the first n evaluations of $\boldsymbol{y}(\cdot)$, i.e., to select $\boldsymbol{x}^{n+1}$ to maximize

$$
I_{PI}(\boldsymbol{x}) = P\left\{\boldsymbol{Y}(\boldsymbol{x}) \not\succeq \boldsymbol{y} \,\forall\, \boldsymbol{y} \in \mathcal{P}^n_{\mathcal{Y}}\right\}. \tag{33}
$$

Equation (33) is termed the probability improvement (PI). The advantage of this criterion is that it is not dependent on the scaling of the output.

A second criterion for selecting $\boldsymbol{x}^{n+1}$ proposed in [12] maximizes the weighted conditional probability

$$I_{CWPI}(\boldsymbol{x}) = P\left\{\boldsymbol{Y}(\boldsymbol{x}) \not\preceq \boldsymbol{y} \,\forall\, \boldsymbol{y} \in \mathcal{P}^n_{\mathcal{Y}}\right\} \tag{34}$$

$$\times \min_{\mathbf{x}_i \in \mathcal{P}^n_{\mathcal{X}}} \sqrt{\sum_{k=1}^{m} \left(\overline{Y}_k(\mathbf{x}) - y_k(\mathbf{x}_i)\right)^2} \tag{35}$$

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

$$\overline{\boldsymbol{Y}}(\mathbf{x}) = \frac{E\left\{\boldsymbol{Y}(\boldsymbol{x}) 1_{\left[\boldsymbol{Y}(\boldsymbol{x}) \not\preceq \boldsymbol{y} \,\forall\, \boldsymbol{y} \in \mathcal{P}^n_{\mathcal{Y}}\right)}\right\}}{P\left\{\boldsymbol{Y}(\boldsymbol{x}) \not\preceq \boldsymbol{y} \,\forall\, \boldsymbol{y} \in \mathcal{P}^n_{\mathcal{Y}}\right\}}$$

(see also [8]). In words $I_{CWPI}(\boldsymbol{x})$ is centroid weighted version of the PI criterion and hence we call it the CWPI criterion. Unlike PI, the relative scaling of the various objectives must be resolved when implementing this method, just as with $EI_{\mathcal{M}}(\boldsymbol{x})$. Using $I_{CWPI}(\boldsymbol{x})$ can be shown to be a generalization of the single-objective expected improvement function.

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 *hypervolume indicator* and the *additive binary-ϵ indicator*. The latter has already been described in Section 4. In the following paragraph a brief description of the former will be given; readers should refer to [18] for an in-depth discussion of Pareto set approximation quality indicators.

The hypervolume indicator of a Pareto front approximation measures the area of the region dominated by this approximation with respect to a given reference point. To calculate the hypervolume indicator of a *finite* set which is a Pareto Front approximation, say B, a reference point, say $\boldsymbol{R}$, must be identified that is weakly dominated by *all* vectors in the output space (see Figure 2). The hypervolume indicator of B is defined to be

$$I_H\left(B, \boldsymbol{R}\right) = \int_{\mathbb{R}^m} 1_{\{\boldsymbol{y} \,|\, \boldsymbol{y} \,\succeq\, \boldsymbol{R},\, B \,\succeq\, \{\boldsymbol{y}\}\}} d\boldsymbol{y}. \tag{36}$$

In words, $I_H\left(B, \boldsymbol{R}\right)$ 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 B so that the larger $I_H\left(B, \boldsymbol{R}\right)$, the better the approximating set B. Figure 2 shows $I_H\left(B, \boldsymbol{R}\right)$ as the shaded area for an $m = 2$ dimensional example with a five point B where $\boldsymbol{R}$ is the upper right-hand corner of the shaded area.

While the hypervolume indicator will be used in this section to compare the effectiveness of the expected maximin improvement criteria with the Pareto

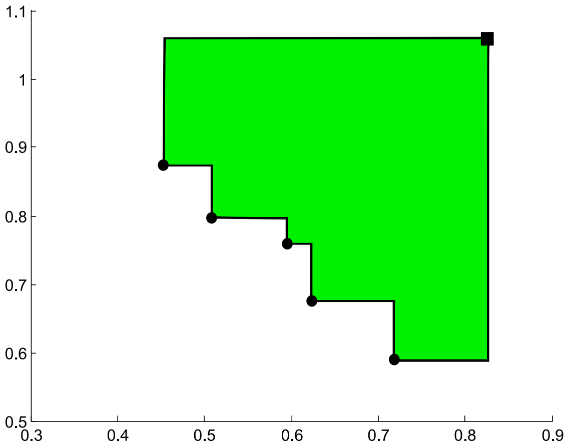

Fig. 2 The filled circles are five-point set B, the filled square is the reference point $\boldsymbol{R}$, and the shaded shaded region is $I_H\left(B, \boldsymbol{R}\right)$

Front approximators presented in [12], it is also possible to use the hypervolume indicator to construct alternative improvement functions (see [6]). Additional remarks about the effectiveness and computational feasibility of such an approach will be discussed in Section 7.

6.1 MOP2 Function

The first example considered in the section is commonly referred to as the MOP2 problem; this test problem was first described in [7]. MOP2 has a $d = 2$-dimensional input space $\mathcal{X} = [-2, 2]^2$, and $m = 2$ objective functions which are

$$y_1(\boldsymbol{x}) = 1 - \exp\left\{-\sum_{i=1}^{2}\left(x_i - \frac{1}{\sqrt{2}}\right)^2\right\} \quad \text{and} \tag{37}$$

$$y_2(\boldsymbol{x}) = 1 - \exp\left\{-\sum_{i=1}^{2}\left(x_i + \frac{1}{\sqrt{2}}\right)^2\right\}. \tag{38}$$

The Pareto set is the line segment

$$\mathcal{P}_{\mathcal{X}} = \left\{\boldsymbol{x} \,:\, x_1 = x_2 \text{ and } -\frac{1}{\sqrt{2}} \leq x_1 \leq \frac{1}{\sqrt{2}}\,,\, -\frac{1}{\sqrt{2}} \leq x_2 \leq \frac{1}{\sqrt{2}}\right\}. \tag{39}$$

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 [8]. The initial

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

	$I_{\epsilon^+}(\mathcal{P}_{\mathcal{Y}}, \mathcal{P}_{\mathcal{Y}}^{10})$			$I_H(\mathcal{P}_{\mathcal{Y}}^{10})$		
Method	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
CWPI-Ind	0.0862	0.0668-0.0927	0.0112	0.2710	0.2649-0.2789	0.0060
PI-Ind	0.1368	0.0937-0.2334	0.0552	0.2531	0.2420-0.2638	0.0096
EMMI-Dep	0.0770	0.0715-0.0882	0.0067	0.2851	0.2811-0.2889	0.0037
CWPI-Dep	0.0937	0.0879-0.0977	0.0041	0.2609	0.2570-0.2647	0.0028
PI-Dep	0.1229	0.0978-0.1608	0.0256	0.2529	0.2306-0.2772	0.0226

design was augmented sequentially with 10 new inputs using EMMI and the two competing methods sketched above. In all three 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 [8], this example utilizes code written by the authors provides more favorable results for PI and CWPI in terms of the hypervolume and additive binary-ϵ indicators.

To compare the Pareto front approximations, both graphical methods and the Pareto set approximation quality indicators were employed. The true Pareto front and various Pareto front approximations were plotted to allow visual inspection of the approximations; the spread of the approximation and its closeness to the 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. Lastly, 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.

From the results in Table 1, it appears that EMMI, calculated using either the independence or dependence GP model, performed significantly better than either the CWPI and PI implementations using either the dependent or independence process model for $(Y_1(\boldsymbol{x}), Y_2(\boldsymbol{x}))$. The area of the dominated hypervolume when using EMMI 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 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 [12]. The plots in Figure 3 show the results for one of the five runs; the spread and uniformity of these points support the superiority of EMMI as well as the numerical measures. While CWPI and PI do not perform poorly, they do not appear as efficient as EMMI because both methods have sequen-

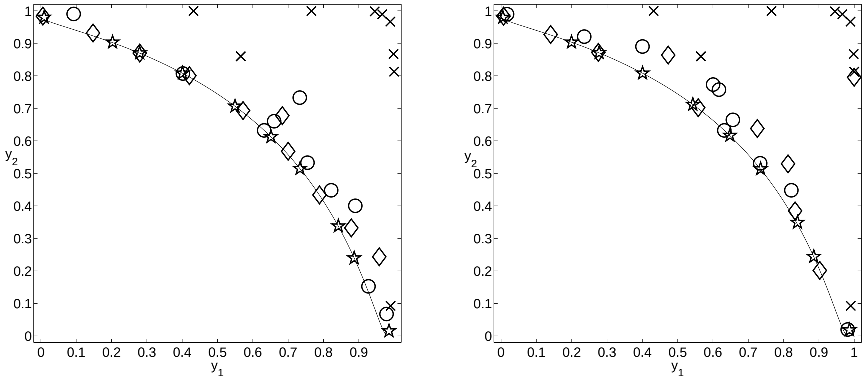

Fig. 3 Sequentially added points using an independence model (left) and dependence model (right) with CWPI (diamonds), EMMI (stars), and PI (circles). The initial 10 outputs are denoted by crosses. The smooth curve running from the top left to the bottom right of each plot is the true Pareto front.

tially 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 sequential add inputs with similar outputs, while CWPI criterion appears to be more effective at spreading out the sequentially added evaluations of the objective function.

The somewhat surprising result here is that the dependence GP model appears to offer little advantage over the less computationally demanding independence model. In fact, on average, it seems to perform slightly worse for almost all improvement criteria, with the lone 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 nonseparable dependence GP model does not model this particular function well. Perhaps a different dependence model for $\boldsymbol{Y}(\boldsymbol{x})$ is more appropriate. Another possible explanation is that the particular dependence model is appropriate, but our estimated covariance parameters, at some stages of the sequential design algorithm, are not globally optimal. Recall that obtaining estimates of the covariance parameters for the dependence model requires maximization of a restricted likelihood function which is a function of seven parameters. This is computationally more difficult than the parameter estimation in the independence model, which requires us to maximize two seperate restricted likelhihood functions, each of which depends on two parameters. It is worth noting that in other examples that the authors have run, which were constructed to satisfy the form of the nonseparable dependence model used here, EMMI with a nonseparable dependence structure produces larger hypervolume indicator and smaller binary-ϵ comparisons with the true Pareto front.

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

	$I_{\epsilon^+}(\mathcal{P}_{\mathcal{Y}}, \mathcal{P}_{\mathcal{Y}}^{40})$			$I_H(\mathcal{P}_{\mathcal{Y}}^{40})$		
Method	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
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
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 [5], 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

$$y_1(\boldsymbol{x}) = (1+g(x_4))\cos\left(\frac{\pi x_1}{2}\right)\cos\left(\frac{\pi x_2}{2}\right)\cos\left(\frac{\pi x_3}{2}\right) \tag{40}$$

$$y_2(\boldsymbol{x}) = (1+g(x_4))\sin\left(\frac{\pi x_3}{2}\right)\cos\left(\frac{\pi x_1}{2}\right)\cos\left(\frac{\pi x_2}{2}\right) \tag{41}$$

$$y_3(\boldsymbol{x}) = (1+g(x_4))\sin\left(\frac{\pi x_2}{2}\right)\cos\left(\frac{\pi x_1}{2}\right) \tag{42}$$

$$y_4(\boldsymbol{x}) = (1+g(x_4))\sin\left(\frac{\pi x_1}{2}\right) \tag{43}$$

where

$$g(x_4) = (x_4 - 0.5)^2 . \tag{44}$$

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

Proceeding in a similar fashion to the MOP2 example, an initial 20 point maximin Latin hypercube design was constructed using the MATLAB function `bestlh` from [8]. Then the original design was augmented sequentially with 20 new points chosen via EMMI, CWPI, and PI. The computations were implemented using the independence and dependence GP models for all three improvement criteria.

In this $m = 4$ dimensional example, graphical methods are problematic to evaluate; thus only the hypervolume indicator $I_H(\mathcal{P}_{\mathcal{Y}}^{40})$ and the additive binary-ϵ indicator $I_{\epsilon^+}(\mathcal{P}_{\mathcal{Y}}, \mathcal{P}_{\mathcal{Y}}^{40})$ were used to compare the various methods. As in the previous example, each algorithm was run five times and the mean, range, and standard deviation of the two quality measures are reported in Table 2.

The DTLZ2 results based on the independence GP model are similar to the MOP2 results. In every run, EMMI-Ind outperforms CWPI-Ind and CWPI-Ind outperforms PI-Ind in terms of both the binary-ϵ indicator and the hypervolume indicator. For the dependence GP model the results differ from those of the MOP2 example. EMMI-Dep has a slightly larger hypervolume indicator than CWPI-Dep on average, but CWPI-Dep has a slightly smaller binary-ϵ indicator than EMMI-Dep on average. The range of both performance measures shows considerable overlap between the two improvement criteria. On the other hand PI-Dep is still performs considerably worse than both EMMI-Dep and CWPI-Dep.

The higher dimensional DTLZ2 example also provides some evidence of the usefulness of the nonseparable dependence GP model. While EMMI-Dep performs considerably poorer with the dependence model in regards to the two Pareto set quality indicators, both CWPI-Dep and PI-Dep seem to have, on average, slightly better performance when using the dependence model. The major downside of the dependence model in this example is that this model 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 four 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 determine 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 $m \geq 3$. Based on the examples presented in the paper and additional ones that are given in the Supplementary Material, the authors recommend using the expected maximin improvement computed using independent Gaussian process models (EMMI-Ind) for problems where it is not possible to supply information concerning possible dependencies among the output functions and where scaling of the objectives can be roughly determined.

We mention one alternative criterion to EMMI-Ind that has attractive performance but is more difficult to implement than EMMI-Ind. This criterion is based on the hypervolume improvement function suggested in [6]. While originally presented as a method of pre-screening inputs in a multiobjective evolutionary algorithm (MOEA), one could use the hypervolume improvement function in the framework presented in Section 5. To do so, define

$$I_{\mathcal{H}}(\mathbf{y}(\boldsymbol{x})) = \begin{cases} 0 & \text{if } \boldsymbol{y}(\boldsymbol{x}) \preceq \mathcal{P}_{\mathcal{Y}}^{n} \\ & \text{or } \boldsymbol{y}(\boldsymbol{x}) \not\preceq \boldsymbol{R} \\ I_H\left(\{\boldsymbol{y}(\boldsymbol{x})\} \cup \mathcal{P}_{\mathcal{Y}}^{n}, \boldsymbol{R}\right) - I_H\left(\mathcal{P}_{\mathcal{Y}}^{n}, \boldsymbol{R}\right) & \text{otherwise.} \end{cases} \tag{45}$$

Then select $\boldsymbol{x}^{n+1}$ to maximize the *expected hypervolume improvement*

$$EI_{\mathcal{H}}(\boldsymbol{x}) = E\left[I_{\mathcal{H}}\left(\mathbf{Y}(\boldsymbol{x})\right) | \boldsymbol{Y}^{m,n} = \boldsymbol{y}^{m,n}\right]. \tag{46}$$

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 MOEA literature that $I_H(\cdot,\cdot)$, and thus $I_{\mathcal{H}}\left(\mathbf{y}(\boldsymbol{x})\right)$, requires considerable computational overhead, even for moderately sized m. Therefore, creating the sample average approximation based on a sample of size S that we suggest for maximizing $EI_{\mathcal{H}}(\boldsymbol{x})$ 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 function 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, 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 $EI_{\mathcal{M}}(\boldsymbol{x})$, $I_{CWPI}(\boldsymbol{x})$, and $EI_{\mathcal{H}}(\boldsymbol{x})$ require scaling each output. In the case of $EI_{\mathcal{M}}(\boldsymbol{x})$, Step 1 of our Section 5 Algorithm creates an empirical scaling that is used with 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 reasonable alternative. Additionally, using either the probability of improvement or the centroid-based expected improvement criteria in conjunction with the dependence GP model shows promise in larger m examples.

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.

Acknowledgements 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.

References

1. Audet, C., Dennis Jr., J.E.: Mesh adaptive direct search algorithms for constrained optimization. SIAM J. on Optimization **17**(1), 188–217 (2006). DOI http://dx.doi.org/10.1137/040603371
2. Balling, R.: The maximin fitness function: A multiobjective city and regional planning. In: C. Fonseca, P. Fleming, E. Zitzler, K. Deb, L. Thiele (eds.) Evolutionary Multi-Criterion Optimization, pp. 1–15. Springer (2003)
3. Banerjee, S., Gelfand, A.E., Finley, A.O., Sang, H.: Gaussian predictive process models for large spatial data sets. Journal Of The Royal Statistical Society Series B **70**(4), 825–848 (2008)
4. Bautista, D.C.: 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 (2009)
5. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multi-Objective Optimization. In: A. Abraham, R. Jain, R. Goldberg (eds.) Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, chap. 6, pp. 105–145. Springer (2005)
6. Emmerich, M.T., Giannakoglou, K.C., Naujoks, B.: Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. IEEE Transactions on Evolutionary Computation **10**(4) (2006)
7. Fonseca, C., Fleming, P.: 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 (1995)
8. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modelling: A Practical Guide. Wiley, Chichester, UK (2008)
9. Fricker, T.E., Oakley, J.E., Urban, N.M.: Multivariate emulators with nonseperable covariance structures. MUCM Technical Report 10/06 (2010)
10. Gelfand, A.E., Schmidt, A.M., Banerjee, S., Sirmans, C.: Nonstationary multivariate process modeling through spatially varying coregionalization. Test (Madrid) **13**, 263–294 (2004)
11. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black–box functions. Journal of Global Optimization **13**, 455–492 (1998)
12. Keane, A.J.: Statistical improvement criteria for use in multiobjective design optimization. AIAA Journal **44**, 879–891 (2006)
13. Knowles, J.: ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjetive optimization problems. IEEE Transactions on Evolutionary Computation **10**(1), 50–66 (2006)
14. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. Statistical Science **4**, 409–423 (1989)
15. Santner, T.J., Williams, B.J., Notz, W.I.: The Design and Analysis of Computer Experiments. Springer Verlag, New York (2003)
16. Schonlau, M.: Computer experiments and global optimization. Ph.D. thesis, Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, CA (1997)
17. Shapiro, A.: Monte Carlo Sampling Methods. In: A. Ruszczynski, A. Shapiro (eds.) Stochastic Programming, *Handbooks in Operations Research and Management Science*, vol. 10, pp. 353 – 425. Elsevier (2003). DOI DOI:10.1016/S0927-0507(03)10006-0
18. Zitzler, E., Knowles, J., Thiele, L.: Quality Assessment of Pareto Set Approximations. In: J. Branke, K. Deb, K. Miettinen, R. Slowinski (eds.) Multiobjective Optimization: Interactive and Evolutionary Approaches, pp. 373–404. Springer (2008)
19. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation **7**(2), 117–132 (2003)