A survey on kriging-based infill algorithms for multiobjective simulation optimization

Peer-reviewed author version

Rojas-Gonzalez, Sebastian & VAN NIEUWENHUYSE, Inneke (2020) A survey on kriging-based infill algorithms for multiobjective simulation optimization. In: COMPUTERS & OPERATIONS RESEARCH, 116 (Art N° 104869).

DOI: 10.1016/j.cor.2019.104869
Handle: http://hdl.handle.net/1942/30505

A survey on kriging-based infill algorithms for multiobjective simulation optimization.

Sebastian Rojas-Gonzalez[a,*], Inneke Van Nieuwenhuyse[a,b]

[a] *Department of Decision Sciences and Information Management, KU Leuven, Belgium.*
[b] *Quantitative Methods, Research Group Logistics, UHasselt, Belgium*

Abstract

This article surveys the most relevant kriging-based infill algorithms for multi-objective simulation optimization. These algorithms perform a sequential search of so-called *infill points*, used to update the kriging metamodel at each iteration. An *infill criterion* helps to balance local exploitation and global exploration during this search by using the information provided by the kriging metamodels. Most research has been done on algorithms for deterministic problem settings; only very recently, algorithms for noisy simulation outputs have been proposed. Yet, none of these algorithms so far incorporates an effective way to deal with heterogeneous noise, which remains a major challenge for future research.

Keywords: Kriging metamodeling, Multiobjective optimization, Simulation optimization, Expected improvement, Infill criteria

1. Introduction

The use of numerical models to simulate and analyze complex real world systems is now commonplace in many scientific and engineering domains (see e.g., Kleijnen (2015), Law (2015) and Rubinstein & Kroese (2016)). Depending on the system under study, and the assumptions of the modeler, the models can be *deterministic* (e.g., in the case of analytical functions) or *stochastic* (e.g., when Monte Carlo simulation or discrete-event simulation is used). Often, the

*Corresponding author
Email address: `sebastian.rojasgonzalez@kuleuven.be` (Sebastian Rojas-Gonzalez)

goal of the modeler is to find the values of controllable parameters (i.e., decision variables) that optimize the performance measure(s) of interest.

The evaluation of the primary numerical models can be computationally expensive; for this reason, different approaches have been developed to provide less expensive *metamodels*, also referred to as *surrogate models*. The use of metamodels allows for a faster analysis than the primary source models, but introduces a new element of error that must be considered in order to allow for valid results and accurate decision making (Meckesheimer et al., 2002). Substantial literature exists on the different metamodeling techniques, such as kriging (Matheron, 1963; Krige, 1951), radial basis functions (Broomhead & Lowe, 1988), polynomial response surface models (Box et al., 1987) and support vector regression (Vapnik, 2013). Metamodeling approaches have become increasingly popular also in the field of multiobjective optimization, in particular in combination with metaheuristics (such as evolutionary algorithms): see, e.g., the recent surveys by Tabatabaei et al. (2015); Diaz-Manriquez et al. (2016); Chugh et al. (2017). The goal of the current article is to survey the state-of-the-art *kriging-based infill* algorithms for multiobjective optimization. This area has not been discussed in detail in the previous surveys, as these took a very broad perspective, allowing different metamodeling approaches. As such, they did not specifically zoom in on kriging-based algorithms, let alone kriging-based infill algorithms.

Kriging metamodels, also referred to as Gaussian Process Regression (GPR) models (Sacks et al., 1989; Rasmussen, 2006) or Gaussian random field models, have been traditionally popular in engineering (see e.g., Forrester et al. (2008); Wang & Shan (2007); Emmerich et al. (2006); Dellino et al. (2007, 2009, 2012)) and machine learning (see e.g., Rasmussen (2006); Koch et al. (2015); Zuluaga et al. (2016)); recently, they have gained increasing popularity also in the Operations Research and Management Science fields (see e.g., Kleijnen (2015); Fu (2014); Picheny et al. (2013)). Kriging metamodels allow for the approximation of outputs (obtained through, e.g., discrete-event simulation) over the entire search space through the kriging predictor (yielding a *global* metamodel); additionally, they quantify the uncertainty of the predictor through the mean square error (MSE), also known as *kriging variance* (Van Beers & Kleijnen, 2003).

In recent years, various multiobjective algorithms have been developed that directly exploit this kriging information (i.e., the predictor *and* its variance) to *sequentially* search the input space for the best input combination(s). We refer to these as *infill algorithms*. Infill algorithms start by simulating a limited set of input combinations (referred to as the initial design), and iteratively select new input combinations to simulate by evaluating an *infill criterion*, also referred to as improvement function or acquisition function (Mockus, 2012), that reflects the kriging information. The kriging metamodel is then sequentially updated with the information obtained from the newly simulated infill points; the procedure repeats until the computational budget is depleted or a desired performance level is reached, and the estimated optimum is returned.

Kriging-based infill algorithms are particularly useful in settings where the computational budget is limited, and the primary simulation model is time-consuming to run: in such settings, they may allow to search the decision space in an efficient way (i.e., limiting the number of simulations to be performed). Yet, there are some downsides too. Evidently, the metamodel outcome is vulnerable to misspecifications in the covariance structure of the random field and/or the covariance parameters, see Rasmussen (2006). The kriging metamodels themselves may be expensive to estimate in settings with a large number of decision variables (Kleijnen, 2015), so their use is primarily advocated in settings with a low-dimensional input space. Optimizing the infill criterion over a continuous domain can be quite challenging, as the criterion function itself contains many local optima (see e.g., Jones (2001); Forrester et al. (2008)). Hence, the analyst requires a heuristic approach, such as a genetic algorithm, to accomplish this. To avoid this issue, the search space is often discretized in articles to evaluate the performance of a newly proposed algorithm, or compare the performance of different methods (see, e.g., Picheny (2015); Feliot et al. (2017)). Enumeration, however, is less suited for problems with high dimensionality, as the number of design points becomes prohibitively high in order to have a sufficient number of high quality solutions (Lemieux, 2009).

We classify the surveyed algorithms as deterministic (i.e., aimed at deterministic problem settings) or stochastic (i.e., aimed at problems with noisy simulation outputs). We do not focus on algorithms that solve specific problems in engineering (such as, e.g., Dellino et al. (2007, 2009, 2012)); rather, we focus on *general purpose* infill algorithms. We distinguish two major categories of infill criteria:

1. Single-objective infill criteria: these have been initially developed for single-objective problems; yet, some multiobjective algorithms continue to use them. The improvement brought by an infill point is measured with respect to each individual objective, or with respect to a scalarized single-objective function.

2. Multiobjective infill criteria: these measure the improvement brought by an infill point with respect to its contribution to the Pareto front. This contribution can be measured using a quality indicator for multiobjective optimizers (e.g., hypervolume), or by evaluating extensions of a single-objective criterion (e.g., multiobjective expected improvement).

The remainder of this article is organized as follows. Section 2 discusses the basics of *kriging*; Section 3 states the most important concepts in multiobjective optimization, Section 4 explains the main types of infill criteria found in the literature, Section 5 focuses on the most relevant kriging-based infill algorithms for deterministic problems, while Section 6 outlines the few infill algoithms for stochastic problems. We conclude the article in Section 7, and identify some promising directions for further research.

2. Kriging metamodeling

Let $f(\mathbf{x})$ be an unknown deterministic function, where $\mathbf{x} = (x_1, ..., x_d)^T$ is a vector of design variables of dimension d. *Kriging* (Sacks et al., 1989; Cressie, 1993), also referred to as *Gaussian process regression* (Rasmussen, 2006; Frazier, 2018), assumes that the unknown response surface can be represented as:

$$f(\mathbf{x}) = \beta + M(\mathbf{x}) \tag{1}$$

where β is a constant trend and $M(\mathbf{x})$ is a realization of a mean zero covariance-stationary Gaussian random field. Instead of a constant trend term β (as in ordinary kriging, see Expression 1), a polynomial trend may also be used (i.e., universal kriging): $\mathbf{f}(\mathbf{x})^T\boldsymbol{\beta}$ where $\mathbf{f}(\mathbf{x})$ then is a vector of known trend functions, and $\boldsymbol{\beta}$ is a vector of unknown parameters of compatible dimension. However, the use of a constant trend term is considered to be preferable (Sacks et al., 1989; Ankenman et al., 2010; Santner et al., 2013; Kleijnen, 2015); all algorithms surveyed in this article use a constant trend for the kriging metamodels.

What relates one observation to another is the *covariance function*, denoted k, also referred to as *kernel*. Multiple covariance functions exist in the field of GPR; the most commonly used are the stationary squared exponential (i.e., the Gaussian kernel, Eq. 2), and Matérn kernel (Eq. 3) (Rasmussen, 2006):

$$k_G(\mathbf{x}^i, \mathbf{x}^j) = \sigma^2 \exp\left[-\sum_{k=1}^{d}\left(\theta_k |x_k^i - x_k^j|\right)^2\right] \tag{2}$$

$$k_{\nu=3/2}(\mathbf{x}^i, \mathbf{x}^j) = \sigma^2 \left[1 + \sqrt{3}\sum_{k=1}^{d}\frac{|x_k^i - x_k^j|}{\theta_k}\right] \times \exp\left[-\sum_{k=1}^{d}\frac{|x_k^i - x_k^j|}{l_k}\right] \tag{3}$$

$$\tag{4}$$

where σ^2, l_k $(k = 1, ..., d)$ are *hyperparameters* that usually need to be estimated, and that denote the process variance, resp. the length-scale of the process along dimension k. Eq. 3 is the Matérn kernel simplified for $\nu = 3/2$, where ν is a hyperparameter that represents the shape (smoothness) of the approximated function (the lower the value of ν, the less smooth the function is). When the hyperparameters are unknown, they are commonly estimated using *maximum likelihood estimation* or *cross validation*. We refer the reader to Santner et al. (2013); Rasmussen (2006); Bachoc (2013) for further discussion of hyperparameter estimation, as these are out of the scope if this survey.

In view of predicting the response at an unsampled point $\mathbf{x}_*$, kriging assumes that the n observations in the vector $\mathbf{y} = f(\mathbf{x})$ can be represented as a sample from a multivariate normal distribution; the conditional probability $P(f(\mathbf{x}_*)|\mathbf{y})$ then represents how likely the response $f(\mathbf{x}_*)$ is, given the observed data (Ebden, 2015):

$$\begin{bmatrix} \mathbf{y} \\ f(\mathbf{x}_*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right) \tag{5}$$

$$P(f(\mathbf{x}_*)|\mathbf{y}) \sim \mathcal{N}(K_* K^{-1}\mathbf{y}, K_{**} - K_* K^{-1} K_*^T) \tag{6}$$

where

$$K = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \tag{7}$$

$$K_* = \begin{bmatrix} k(\mathbf{x}_*, \mathbf{x}_1) & k(\mathbf{x}_*, \mathbf{x}_2) & \dots & k(\mathbf{x}_*, \mathbf{x}_n) \end{bmatrix} \tag{8}$$

$$K_{**} = \begin{bmatrix} k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \tag{9}$$

Consequently, the best estimate for $f(\mathbf{x}_*)$ is the mean of this distribution (Eq. 10), and the uncertainty of the estimate is given by the kriging variance (Eq. 11):

$$\bar{f}(\mathbf{x}_*) = K_* K^{-1}\mathbf{y} \tag{10}$$

$$\mathrm{var}(f(\mathbf{x}_*)) = K_{**} - K_* K^{-1} K_*^T \tag{11}$$

3. Multiobjective optimization

This section briefly explains the important concepts and terminology in multiobjective optimization (section 3.1), as well as the performance evaluation of deterministic multiobjective optimizers and additional considerations for performance evaluation in stochastic settings (section 3.2).

3.1. Concepts and terminology

In general, a multiobjective optimization (hereafter referred to as MO) problem can be formulated as follows (Deb et al., 2002): $\min[f_1(\mathbf{x}), ..., f_m(\mathbf{x})]$ for m objectives and a vector of decision variables $\mathbf{x} = [x_1, ..., x_d]^T$ in the decision space D (usually $D \subset \mathbb{R}^d$), with $f : D \to \mathbb{R}^m$ the vector-valued function with coordinates $f_1, ..., f_m$ in the objective space $\Theta \subset R^m$.

Usually, there are tradeoffs between the different objectives; the goal then is to find a set F of all vectors $\mathbf{x}^* = [x_1^*, ..., x_d^*]^T$ where one objective cannot be improved without negatively affecting any other objective. The points in this solution set are referred to as *non-dominated* or *Pareto-optimal* points, and form the *Pareto set* (see definition 3.1 for a formal definition of the concept of (strict) dominance; throughout this survey we assume that all objectives have to be minimized).

Definition 3.1. *For* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *two vectors in* D (Zitzler et al., 2003):

- $\mathbf{x}_1 \prec \mathbf{x}_2$ means $\mathbf{x}_1$ dominates $\mathbf{x}_2$ iff $f_j(\mathbf{x}_1) \leq f_j(\mathbf{x}_2), \forall j \in \{1, .., m\}$, and $\exists j \in \{1, .., m\}$ such that $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$

- $\mathbf{x}_1 \prec\prec \mathbf{x}_2$ means $\mathbf{x}_1$ strictly dominates $\mathbf{x}_2$ iff $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2), \forall j \in \{1, .., m\}$

The evaluation of these solutions in the objective space corresponds to the *Pareto front*, denoted $\mathcal{P}^\Theta$. Mathematically, all Pareto-optimal points are equally acceptable solutions (Miettinen, 1999); the final solution preferred by the decision maker then depends on his/her preferences. A common approach to search for Pareto-optimal points is to *scalarize* the objectives into one performance function, by assigning weights (preferences) to each objective. By varying the set of weight values uniformly, we can obtain points that fall between the objectives' extremes, and thus construct the Pareto front (Das & Dennis, 1997).

Numerous scalarization functions/methods have been put forward in the literature, and the choice depends mainly on the geometrical properties of the problem (Miettinen, 1999). The following functions, and their variations, are most commonly used (Miettinen & Mäkelä, 2002):

1. Weighted Tchebycheff scalarization function:

$$\max_{j=1,..,m} \lambda_j (f_j(\mathbf{x}) - z_j^*) \tag{12}$$

2. Augmented Tchebycheff scalarization function:

$$\max_{j=1,..,m} \lambda_j (f_j(\mathbf{x}) - z_j^*) + \rho \sum_{j=1}^{m} \lambda_j f_j(\mathbf{x}) \tag{13}$$

3. Weighted sum scalarization function:

$$\sum_{j=1}^{m} \lambda_j f_j(\mathbf{x}) \tag{14}$$

with $\lambda_j \geq 0$, $\sum_{j=1}^{m} \lambda_j = 1$, $\forall j \in \{1, .., m\}$. The result is thus a single-objective problem. In equations 12 and 13, z_j^* represents the *ideal* value for objective j, and thus provides a lower bound for each objective function in the Pareto set; and ρ is a small positive value.

Literature is extensive in the field of multiobjective optimization, with important advances in metaheuristic-based methods (see, e.g., Abraham et al. (2005); Zhou et al. (2011); Emmerich & Deutz (2018), analytical methods (see, e.g., Miettinen (1999); Giagkiozis & Fleming (2015)), and *scalarization methods*, which are commonly used in the literature to transform the problem into a single-objective problem (see e.g., Miettinen & Mäkelä (2002); Knowles (2006)). The interested reader is referred to Marler & Arora (2004) for a comprehensive survey on MO in deterministic engineering problems, and to Gutjahr & Pichler (2016) for a recent review on non-scalarizing methods for stochastic MO.

3.2. Performance evaluation of multiobjective optimizers

Measuring the quality of such a Pareto front approximation is a non-trivial task (Zitzler et al., 2002), as the so-called "true" Pareto front is usually unknown. Intuitively, a *good* Pareto front is characterized by *richness* (i.e., the Pareto front needs to be well populated) and *diversity* (i.e., the Pareto optimal points should be well spread with respect to all the objectives).

Numerous quantitative performance indicators have been developed for assessing the quality of the Pareto front in *deterministic* problem settings (see Riquelme et al. (2015) for a recent review); some of the most widely used quality indicators are the *hypervolume* (Zitzler et al., 2007), the *inverted generational distance* (Coello & Sierra, 2004), and the R indicator family (Hansen & Jaszkiewicz, 1998). The hypervolume is particulary popular, as it is the only indicator that is *strictly monotonic* (i.e., an increase in the hypervolume value immediately implies an improvement in the Pareto front approximation). However, the runtime complexity of the hypervolume is exponential in the number of objectives (Bader & Zitzler, 2011).

In multiobjective *stochastic* simulation optimization, the problem is more complex as the objectives are not only in conflict, but also perturbed by noise. In general, relying on the *observed* mean objective values to determine the non-dominated points (as in Definition 3.1) may lead to two possible errors due to sampling variability: designs that actually belong to the non-dominated set can be wrongly considered dominated, or vice versa. The algorithm needs to take into account the noise disturbing the observations during the optimization process, otherwise the model may lead to incorrect inference about the system's performance (see, e.g., Knowles et al. (2009), who applied ParEGO to noisy problems, showing the detrimental effect of the noise on the results).

The most commonly used method for handling noise during optimization is to evaluate the same point a number of times and use the mean of these replications as the response value. However, when the noise is high and/or strongly heterogeneous, this method may fail to provide accurate approximations with limited computational budget (Jin & Branke, 2005). It is thus necessary to use more advanced procedures that aim to correctly identify the systems with the true best expected performance, such as *dynamic resampling*, *probabilistic dominance* or *multiobjective ranking and selection* (MORS).

In Syberfeldt et al. (2010), the authors propose to dynamically vary the additional number of samples based on the estimated variance of the observed objectives' values. The technique, called confidence-based dynamic resampling, allows for the assessment of the observed responses at a particular confidence level before determining determining dominance, and aims to avoid unnecessary resampling (i.e., when it provides little benefit). Another example is the RTEA algorithm of Fieldsend & Everson (2015). Instead of using variance learning techniques, it is done during the evolutionary phase of the algorithm, by tracking the improvement on the Pareto set (as opposed to the Pareto front). Their algorithm focuses on the observation that the best estimate for the noise-free objectives associated with a design improves with the number of samples taken.

Another approach is to use the concept of *probabilistic dominance*: the probability that one solution dominates another needs to be higher than some specified degree of confidence to determine domination (Fieldsend & Everson, 2005). For example, da Fonseca et al. (2001) (see also Zitzler et al. (2008)) propose to use the *expected* values of any deterministic indicator to compare the quality of different Pareto fronts with a certain confidence level, using non-parametric statistical tests. Similarly, in Gong et al. (2010), the probabilistic dominance is defined by comparing the volume in the objective space enclosed by a given point using confidence intervals, and uses the center point of these volumes to determine the dominance relationship. In Basseur & Zitzler (2006), each solution is inherently associated with a probability distribution over the objective space; a probabilistic model that combines quality indicators and uncertainty is created and then used to calculate the expected value for each solution. Another approach is presented in Trautmann et al. (2009) and Voß et al. (2010), where Pareto dominance is defined using the standard deviations of the observed mean approximations: the standard deviation is added to the mean such that dominance is defined with the worst case objective values.

A more advanced alternative is to use MORS methods; these, however, are very scarce in the literature (Hunter et al., 2019). MORS procedures aim to ensure a high probability of *correctly* selecting a non-dominated design, by smartly distributing the available computational budget between the search of infill points and replicating on critically competitive designs, in order to achieve sufficient accuracy. Analogously, they avoid spending budget on those designs that are clearly dominated and are, thus, not interesting to the decision-maker. Some of the most relevant works in MORS include Lee et al. (2010), Bonnel & Collonge (2014, 2015), Li et al. (2015), Feldman et al. (2015) and Branke et al. (2016), but substantial work remains to be done in this regard.

4. Infill criteria

As mentioned in the Introduction, the infill criterion is a key concept for any kriging-based algorithm: it estimates the *improvement* brought by each given non-simulated point to the solution of the problem by exploiting the metamodel information. Substantial research has been done on infill criteria for deterministic single and multiobjective problems (see e.g., Jones (2001); Wagner et al. (2010); Parr et al. (2012)); we refer the interested reader to Hoffman et al. (2011); Brochu et al. (2010) for how to select an infill criterion.

In this survey, we categorize papers based on the type of infill criterion used. We distinguish between single-objective infill criteria and multi-objective infill criteria. *Single-objective* infill criteria are also used in multi-objective infill algorithms, either when the multiple objectives are scalarized into (one) objective function (which basically reduces the MO problem to a single-objective problem), *or* when the improvement is being measured for each objective function separately and used to determine the dominance relationship between the points. *Multi-objective infill criteria*, in contrast, measure the contribution of the infill point with respect to the Pareto front (e.g., by looking at the hypervolume improvement brought by that point), or they consider an extension of a single-objective infill criterion.

4.1. Single-objective infill criteria

We mainly distinguish six types of criteria in the literature:

1. Mean and variance values (MI): The prediction values and uncertainties provided by the kriging metamodels are used directly in the search phase of the algorithms (Emmerich et al., 2006).

2. Expected improvement (EI): The EI measures the expected value of improvement relative to the currently found minimum goal value f_{min} at a certain point $\mathbf{x}$, in view of improving the balance between local exploitation and global exploration of the kriging metamodel:

$$E[I(\mathbf{x})] = (f_{min} - \hat{f}(\mathbf{x}))\Phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{\hat{s}}\right) + \hat{s}\phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{\hat{s}}\right) \quad (15)$$

   where $\Phi(\cdot)$ denotes the normal cumulative distribution, ϕ denotes the normal probability density function, and $\hat{f}(\mathbf{x})$ and $\hat{s}$ respectively refer to the predicted response and standard deviation. The EI was popularized through the well-known Efficient Global Optimization (EGO) algorithm (Jones et al., 1998), developed for deterministic single-objective black-box optimization problems. At each iteration, the EGO algorithm selects the solution that maximizes EI as the infill point. The pros and cons of the EI have been extensively studied (see Ponweiser et al. (2008b); Santner et al. (2013) for further details).

3. Probability of improvement (PoI): PoI is defined as the probability that the output at $\mathbf{x}$ is at or below a target value T (with $T \leq f_{min}$, Ulmer et al. (2003)):

$$P[I(\mathbf{x})] = \Phi\left(\frac{T - \hat{f}(\mathbf{x})}{\hat{s}}\right) \quad (16)$$

   where $\Phi(\cdot)$ denotes the standard normal cumulative distribution, and $\hat{f}(\mathbf{x})$

and $\hat{s}$ again refer to the predicted response and standard deviation respectively. Areas with high PoI are more promising to explore. We refer to Jones (2001) and Mockus (2012) for further details on the probability of improvement.

4. Probability of feasibility (PoF): The PoF is used when expensive constraint functions are present (Forrester et al., 2008; Singh et al., 2014). It measures the degree to which a sample satisfies the constraints; thus, it is normally used in conjunction with the PoI or EI. Let $\hat{g}(\mathbf{x})$ be the constraint function prediction and $\hat{s}^2(\mathbf{x})$ the prediction variance. Then the probability that a constraint is met, i.e., the probability of the prediction being greater than the constraint limit, $P(G(\mathbf{x}) > g_{min})$, is defined as (Forrester & Keane, 2009):

$$P(G(\mathbf{x}) > g_{min}) = \Phi\left(\frac{\hat{g}(\mathbf{x}) - g_{min}}{\hat{s}_i(\mathbf{x})}\right) \tag{17}$$

where Φ is the standard normal cumulative distribution, g_{min} the bound for the constraint value, and $G(\mathbf{x})$ a is the normally distributed random variable with mean $\hat{g}(\mathbf{x})$ and variance $\hat{s}^2(\mathbf{x})$. For several expensive constraint functions modeled using kriging, the combined PoF is given by the product of all the individual probabilities (Singh et al., 2014).

5. Lower confidence bound (LCB): The goal of the LCB is to increase the number of evaluations in promising regions in the design space that haven't been explored yet, by directing the search using a user-defined confidence bound of the approximated response:

$$f_{lb}(\mathbf{x}) = \hat{f}(\mathbf{x}) - \omega\hat{s} \tag{18}$$

where $\omega \in [0, 3]$. By varying the value of ω, the user can focus the search on local areas or explore the design space more globally (Emmerich et al., 2006). We refer to MacKay (1998) and Auer (2002) for more discussion on the lower (minimization) and upper (maximization) confidence bounds.

6. Entropy search (ES): An entropy-based search seeks to minimize the uncertainty in the *location* of the optimal value (Barber, 2012). As discussed in Section 2, we are interested in the conditional probability $P(f(\mathbf{x}_*)|\mathbf{y})$ (i.e., how likely the response of a new point $\mathbf{x}_*$ is, given the observed data $\mathbf{y} = f(\mathbf{x})$). An entropy-based criterion seeks for (infill) points that minimize the entropy H of the induced distribution $P(f(\mathbf{x}_*)|\mathbf{y})$. Derivation of entropy-based criteria is non-trivial and several assumptions on the nature of the distribution must be made (Barber, 2012) (see also Hernández L. et al. (2014) and Hennig & Schuler (2012)).

4.2. Multiobjective infill criteria

Using scalarization, in principle, any infill criterion developed for single-objective simulation optimization can be used to search and select candidate points. However, a disadvantage of the scalarization approach is that without further assumptions (e.g., convexity) on the objectives, some Pareto-optimal solutions may not be detected (Boyd & Vandenberghe, 2004). Fortunately, there has been important progress in developing multiobjective expected improvement criteria, where instead of measuring the improvement of each individual (or scalarized) objective, the improvement is an estimate of the *progress* brought by a new sampled point to the set of non-dominated points. We distinguish two different types of multiobjective criteria in the literature:

1. Indicator-based: These approaches use quantitative performance indicators as infill criteria, reflecting how much the quality indicator improves if the corresponding individual is added to the current Pareto front (Zitzler & Künzli, 2004). A specific quality indicator may be directly used to assign a fitness function to each solution (such as in Ponweiser et al. (2008a), which uses the hypervolume contributions). Alternatively, one estimates the expected improvement in the quality indicator for each solution, such as in Emmerich et al. (2006, 2011), which use the expected hypervolume improvement (EHI), or Couckuyt et al. (2014) who uses EHI

and hypervolume-based PoI. For constrained problems, the EHI is usually combined with the multiobjective PoF (e.g., Martinez F. & Herrero P. (2016); Feliot et al. (2017)).

2. Extensions of single objective criteria: These approaches devise closed-form extensions to the single-objective criteria; examples are the Maximin EI (Svenson & Santner, 2016), Euclidean-based EI (Keane, 2006; Forrester et al., 2008), multiobjective PoI and ES (Picheny, 2015), and Desirability-based EI (Henkenjohann et al., 2005, 2007). In Chugh et al. (2016), the MI values for each objective are used in combination with the so-called angle penalized distance (APD) to select infill points.

Further details on these criteria and their respective algorithms are discussed in Section 5.2.

5. Kriging-based multiobjective infill algorithms for deterministic problems

This section discusses infill algorithms developed for deterministic MO problems. Section 5.1 focuses on algorithms using single-objective infill criteria (see section 4.1), while section 5.2 discusses algorithms that apply multiobjective infill criteria (as discussed in section 4.2).

5.1. Algorithms with single-objective infill criteria

The multiobjective kriging-based optimization algorithms surveyed in this section are summarized in Table 1. As illustrated in Figure 1, to search for infill points, they either scalarize the objectives into one before fitting a (single) kriging model, or they fit separate models to each individual objective. In the latter case, the improvement is measured with respect to each separate objective, but the selection of infill points is based on the optimal tradeoff between the objectives (i.e., a non-dominated sort is run based on the metamodel predictions).

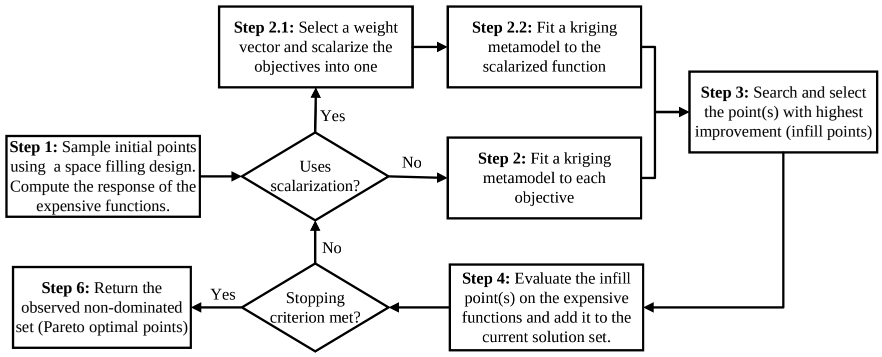

Figure 1: Generic structure of a kriging-based MO algorithm with single-objective infill criterion.

As is common in kriging-based sequential algorithms, a latin hypercube sample (LHS) is used for the initial design in the first step. Jones et al. (1998) suggests to fix the number of initial design points to $11d - 1$, with d the dimension of the search space. In further works, such as Jones (2001); Knowles (2006), the number of points is recommended to be at least 10 times the number of dimensions, based on extensive empirical knowledge. In the second step, before fitting one or several metamodels, the objectives are normalized with respect to their known (or estimated) ranges so that each objective function lies between $[0, 1]$. Step 3 selects a point or a set of points with highest improvement (all algorithms in Table 1 use a genetic algorithm to that end); this infill point(s) are then evaluated using the expensive simulator in Step 4, after which the kriging model is updated with the new information, unless a stopping criterion is met.

One of the first works extending EGO for MO of deterministic problems is the Multi-EGO algorithm of Jeong & Obayashi (2005). Multi-EGO exploits the advantages of the EI criterion for each of the objectives in the search of infill points. For a given population, the EIs for each objective are used to determine the non-dominated points, as opposed to using the kriging predictions directly. This means not necessarily the points the maximize the EI for each objective will be selected, as in the original EGO algorithm of Jones et al. (1998), but those with the optimal EI tradeoffs. The algorithm is evaluated on

Algorithm	Reference	Uses scalarization	Infill criterion	Search space	Numerical experiments
Multi-EGO	Jeong & Obayashi (2005)	No	EI	Continuous	Problem: practical Decision variables: 26 Objectives: 2 Constraints: Yes
ParEGO	Knowles (2006)	Yes Eq. 13	EI	Continuous	Problem: analytical Decision variables: 2 $\sim$ 8 Objectives: 2 $\sim$ 3 Constraints: No
MOEA/D-EGO	Zhang et al. (2010)	Yes Eq. 12 and 14	EI	Continuous	Problem: analytical Decision variables: 2 $\sim$ 8 Objectives: 2 $\sim$ 3 Constraints: No
KEEP	Davins-Valldaura et al. (2017)	Yes Eq. 13	EI	Continuous	Problem: practical/analytical Decision variables: 12 Objectives: 2 Constraints: No
K-MOGA KD-MOGA	Li et al. (2008) Li et al. (2009)	No	MI and LCB	Continuous and discretized	Problem: practical/analytical Decision variables: 2 $\sim$ 5 Objectives: 2 Constraints: Yes

Table 1: Overview of deterministic single-objective infill algorithms.

a biobjective engineering problem in the field of aerodynamic design, showing promising results.

ParEGO (Knowles, 2006) and MOEA/D-EGO (Zhang et al., 2010) have become two popular algorithms that employ a kriging metamodel in the optimization framework in order to speed up computations. In both cases, a scalarization function is used to aggregate the multiple criteria into one. The key difference between both approaches is that ParEGO optimizes the EI value of one single-objective subproblem per iteration, and thus can generate only one infill point to evaluate at each generation. By contrast, MOEA/D-EGO considers multiple scalarized subproblems simultaneously (based on the former algorithm MOEA/D of Zhang & Li (2007)), and thus produces several infill points in each iteration (see also Liu et al. (2007) for one of the first works that extended MOEA/D using GRF metamodels). Both algorithms use the EI criterion as defined in Jones et al. (1998) (see Equation 15).

Knowles (2006) finds ParEGO to perform well on a series of benchmark problems with maximum 3 objectives and 8 decision variables. The hypervolume and epsilon indicators of the ParEGO solutions are compared against the performance of the famous NSGA-II non-surrogate-assisted evolutionary algorithm (Deb et al., 2002), showing that ParEGO explores the objective space more efficiently, yielding better results than NSGA-II with a limited number of evaluations. However, NSGA-II outperforms ParEGO in some problems with high dimensionality; according to the authors of ParEGO, a sparser initial design for higher dimensions (6-8 decision variables) may be worth considering to improve its performance.

MOEA/D-EGO's performance is evaluated in Zhang et al. (2010) against ParEGO and SMS-EGO (Ponweiser et al. (2008a), discussed in Section 5.2). The experimental study on several benchmark problems (see e.g., Huband et al. (2006)) showed that when the number of function evaluations allowed is limited, the performance of MOEA/D-EGO is at least as good as ParEGO and SMS-EGO. However, due to the parallel optimization of several scalarized functions in one iteration, MOEA/D-EGO has the advantage of proposing several infill points per iteration. This makes it more suitable for solving multiobjective problems in practice, as convergence to a front is faster than sampling a single point per iteration (Zhang et al., 2010).

A recent extension of the ParEGO algorithm is presented in Davins-Valldaura et al. (2017), where the authors argue that ParEGO tends to favor solutions suitable for the reduction of the surrogate model error, rather than for finding the best possible non-dominated solutions. The main feature of their proposed algorithm, referred to as KEEP (Kriging for Expensive Evaluation Pareto), is to enhance the convergence speed and thus to reduce the total number of function evaluations by means of a so-called *double kriging strategy.* A closed form of a modified version of the EI is presented, that jointly accounts for the objective function approximation error and the probability to find Pareto Set solutions. The proposed infill criterion uses the information of both kriging metamodels, where the first one is obtained as in ParEGO (steps 1-3 in Figure 1), in order to select the best infill point, whereas the second model aims to rapidly locate areas in the decision space with high probability of containing Pareto-optimal points. Experimental results on benchmark multiobjective functions show a small improvement in the hypervolume indicator values of KEEP with respect to ParEGO and other non-kriging-assisted evolutionary multiobjective algorithms.

Li et al. (2008) presents a kriging-based multiobjective genetic algorithm (K-MOGA), where the kriging variance is exploited as a measure of correctness of the predicted responses. At each generation, a kriging model is fitted to each objective and used to evaluate each point in the population. If the kriging variance (i.e., the prediction uncertainty) is higher than some defined threshold for any point in this population, the primary expensive simulation model is used on that point to yield the true response values. This way the algorithm only computes the expensive responses when the uncertainty of the predictor is high. Closed forms for the threshold criteria are devised for the objective functions and constraints, if the latter are present. Using the true or approximated responses for all the points in the population, a non-dominated sort is used to determine the non-dominated points (i.e., the parents for the next evolutionary phase).

K-MOGA is compared against the performance of the non-kriging-based version (MOGA) on several test functions. The results show that K-MOGA is able to achieve comparable convergence and diversity of the Pareto frontier with a substantial reduction of the computational effort relative to MOGA. The authors present an improvement to K-MOGA in Li et al. (2009), using an adaptive space-filling design (DOE) in each generation, in order to sample better points during reproduction. The authors conclude that the algorithm, referred to as KD-MOGA (kriging-DOE-MOGA), performs better than MOGA and K-MOGA on several test functions.

A study presented in Voutchkov & Keane (2010) examines the use of MI and EI compared with different search strategies, also including pure random search. Experiments on the ZDT test functions reinforce the well-known result that the EI criterion performs best overall. The authors also observe that for high-dimensional problems (e.g., with 25 decision variables), surrogate-based strategies don't perform as well as with e.g., 10 dimensions. In such cases, combinations with other techniques, such as genetic algorithms, are necessary during the search phase of the algorithms.

5.2. Algorithms with multiobjective infill criteria

These approaches should balance the quality of the Pareto-front approximation and the improvement of the global model quality. They use the kriging metamodels to compute an approximation of the responses for all the points in the search space, and these are evaluated in the multiobjective criterion to yield the best infill point(s). Depending on the algorithm, one or several points can be selected at the end of each iteration. As stated in the Introduction, we only consider algorithms where the kriging variance is exploited during optimization. Depending on the nature of the infill criterion used, evaluating the improvement of every point may incur very high computational costs due to multivariate piecewise integrations (Couckuyt et al., 2014). Figure 2 shows the general steps followed by these algorithms; Table 2 summarizes the surveyed algorithms.

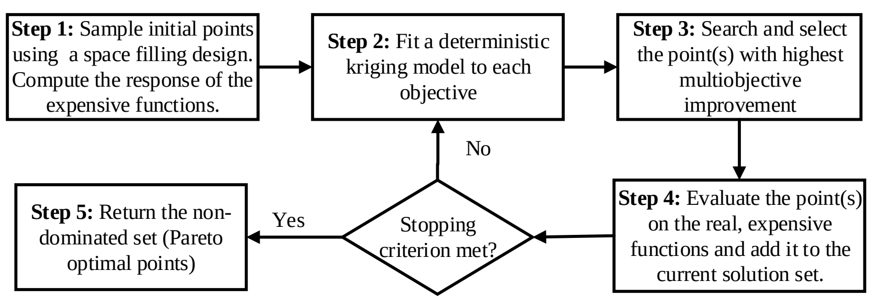

Figure 2: Generic structure of a kriging-based MO algorithm with multiobjective infill criterion.

Methods that employ multiobjective infill criteria normally assume that each of the objective functions $f_j(\mathbf{x}), \forall j \in \{1, .., m\}$ is a sample path of a random field M_j (see Eq. 1), and that the responses are independent (Wagner et al., 2010). Though it is possible to account for correlation between the multiple objectives, for instance by using co-kriging models (see Kleijnen & Mehdad (2014)), recent research shows that such models are more complex and don't significantly outperform independent models in the search for solutions (Fricker et al., 2013).

Algorithm	References	Infill criterion	Search space	Computational cost	Numerical experiments
SExI-EGO	Emmerich et al. (2006) Emmerich et al. (2011)	EHI	Continuous	High	Problem: analytical Decision variables: $2 \sim 10$ Objectives: 2 Constraints: No
SMS-EGO	Ponweiser et al. (2008a) Emmerich et al. (2006)	LCB and EHI	Continuous	High	Problem: analytical Decision variables: $3 \sim 6$ Objectives: $2 \sim 5$ Constraints: No
EMO	Couckuyt et al. (2012) Couckuyt et al. (2014)	EHI and Hypervolume-based PoI	Continuous	Low	Problem: analytical Decision variables: 6 Objectives: $3 \sim 6$ Constraints: No
ECMO	Couckuyt et al. (2014) Singh et al. (2014)	Hypervolume-based PoI and PoF	Continuous	Low	Problem: practical/analytical Decision variables: $2 \sim 3$ Objectives: $2 \sim 7$ Constraints: Yes
MEI-SPOT	Keane (2006) Forrester et al. (2008)	Euclidean-based EI and PoI	Continuous	High	Problem: practical Decision variables: 2 Objectives: 2 Constraints: No
KEMOCO	Martinez F. & Herrero P. (2016)	EHI and PoF	Discretized	High	Problem: analytical Decision variables: 2 Objectives: 2 Constraints: Yes
BMOO	Feliot et al. (2017)	EHI and PoF	Discretized	Low	Problem: analytical Decision variables: $2 \sim 6$ Objectives: $2 \sim 5$ Constraints: Yes
Multi-EI	Henkenjohann et al. (2005) Henkenjohann et al. (2007)	Desirability-based EI	Discretized	High	Problem: practical Decision variables: 3 Objectives: 3 Constraints: No
SUR	Picheny (2015)	PoI and ES	Discretized and continuous	High	Problem: practical/analytical Decision variables: $1 \sim 6$ Objectives: 2 Constraints: No
EMmI	Svenson & Santner (2016) Bautista (2009)	Maximin EI	Discretized	Low	Problem: analytical Decision variables: $2 \sim 4$ Objectives: $2 \sim 4$ Constraints: No
K-RVEA	Chugh et al. (2016)	MI and APD	Continuous	Low	Problem: analytical Decision variables: 10 Objectives: $3 \sim 10$ Constraints: No

Table 2: Summary of multiobjective infill criteria and related algorithms.

The hypervolume (i.e., the volume of points in objective space between the Pareto front and a reference point) is commonly used in indicator-based algorithms (Zitzler et al., 2008). The best improvement is obtained with the point that maximizes this hypervolume. A drawback with this method is that the indicator is computationally expensive due to piecewise integrations, so its evaluation becomes infeasible if the problem dimensionality is large (Emmerich et al., 2011; Auger et al., 2012). One of the early works that considered a hypervolume-based search in multiobjective optimization assisted by kriging metamodels appears in Emmerich et al. (2006). In this work, the most commonly used infill criteria (i.e., MI, EI, PoI and LCB) are analyzed in detail for both the single- and bi-objective cases. The EI criterion performed best in terms of accuracy for the single-objective case, while MI performed badly. Thus, the authors propose to formalize the EI as a multiobjective infill criterion using hypervolume as a fitness indicator. The resulting criterion is referred to as Expected Hypervolume Improvement (EHI), and its calculation requires integrating the improvement function over the entire non-dominated region A as:

$$\mathrm{EHI}(\mathbf{x}) = \int_{\mathcal{P}\in A} I(\mathcal{P}) \prod_{j=1}^{m} \frac{1}{\hat{s}_j} \phi\left(\frac{y_j - \hat{y}_j}{\hat{s}_j}\right) dy_j \tag{19}$$

where $I(\mathcal{P})$ is the improvement function (i.e., hypervolume) of a given Pareto front.

The EHI is used in the SExI-EGO algorithm later proposed by Emmerich et al. (2011). To speed up computations, the authors propose to divide the response space in a series of cells, such that the response value of a given $\mathbf{x}$ has an associated probability of belonging to a non-dominated cell. This, however, requires the algorithm to iterate over the total number of cells, which in turn grows exponentially with the number of objectives. There has been significant progress in increasing the speed of the EHI calculation (see e.g., Couckuyt et al. (2014); Hupkens et al. (2014); Zhan et al. (2017)); nevertheless, it has been claimed that EHI is feasible for 3 objectives at most (Hernández L. et al., 2016). The performance of SExI-EGO was recently compared against similar algorithms in Zaefferer et al. (2013) and Shimoyama et al. (2013), where it's shown to be efficient in the search of solutions for unconstrained problems, but at a high computational cost. Its performance is significantly reduced for problems with constraints.

A similar idea to the SExI-EGO algorithm was earlier developed in Ponweiser et al. (2008a). The algorithm, referred to as SMS-EGO (S-Metric Selection EGO), uses the hypervolume improvement as an infill criterion, and the search is based on the LCB. The kriging responses are stored in vectors as $\hat{\mathbf{y}}_{pot} = \hat{\mathbf{y}} - \alpha\hat{\mathbf{s}}$, where $\hat{\mathbf{y}}_{pot}$ is the vector containing the *lower confidence bounds* of the predicted outputs, for some constant α (see Equation 18). The hypervolume contribution is computed for all (non-dominated) points at each iteration; the best point is selected and added to the overall solution set to update the kriging metamodel. Experimental results in Ponweiser et al. (2008a) show that SMS-EGO outperforms ParEGO and Multi-EGO in terms of quality of the Pareto front; yet, as shown in the experiments of Zhang et al. (2010) and Chugh et al. (2016), the computational cost of SMS-EGO is quite high, as it evaluates the (expensive) hypervolume indicator at all potential members of the Pareto front.

An alternative approach to using quality indicators, is to derive an extension of a single-objective criterion, such as EI or PoI, to multiobjective settings. Keane (2006) (see also Forrester et al. (2008)) derive a multiobjective EI criterion using the Euclidean distance between a given objective vector and its nearest non-dominated point, and the probability that the new point is not dominated by any point in the current front. Thus, the corresponding algorithm, referred to as MEI-SPOT in the literature, only selects infill points that dominate current Pareto-optimal points. Closed form expressions of the criterion are devised for the biobjective case. Moreover, the computational cost of this criterion grows exponentially with the number of objectives (Keane, 2006).

The experiments carried out in Wagner et al. (2010) and Zaefferer et al. (2013) compare MEI-SPOT with other similar approaches, such as SExI-EGO, SMS-EGO and MSPOT (Zaefferer et al., 2013). These algorithms where tested under the same conditions (i.e., bi-dimensional decision and objective space, one infill point sampled per iteration and maximum 80 evaluations). The results show that no particular algorithm performs best in terms of quality of solutions, with the exception of MEI-SPOT, which performed significantly worse than the rest.

Couckuyt et al. (2012) and Couckuyt et al. (2014) propose more efficient methods to calculate the multiobjective Euclidean-based EI and hypervolume-based PoI, as well as a fast method to calculate the EHI. An algorithm is developed to evaluate the efficiency of these infill criteria, referred to as Efficient Multiobjective Optimization (EMO). The proposed methods seem to be among the most competitive in the literature for the calculation of these criteria, as shown in the experimental results. The performance of EMO is at least as good as the performance of state-of-the-art evolutionary multiobjective algorithms, for benchmark problems having up to 6 objectives. Moreover, the proposed EHI criterion delivers competitive results for a significantly lower cost. Furthermore, an extension of the EMO algorithm is proposed in Singh et al. (2014) which considers expensive constraints, referred to as ECMO (Efficient Constrained Multiobjective Optimization). The key contribution of ECMO is to combine a criterion for improvement of the current Pareto front (i.e., hypervolume-based PoI), and a criterion for only considering feasible solutions (i.e., PoF). The proposed algorithm outperforms the (non-kriging-based) NSGAII (Deb et al., 2002) for up to 7 objectives with expensive constraints.

Analogous to ECMO, the Kriging-based Efficient Multi-Objective Constrained Optimization (KEMOCO) algorithm developed in Martinez F. & Herrero P. (2016) also considers fitting kriging metamodels to expensive constraints, and combines the EHI with the PoF to search for infill points. The proposed sequential procedure is divided in two phases. The first one is used to generate an initial feasible approximation of the Pareto front by sampling points in regions of the design space with high PoF. When a user-defined target number of feasible designs is reached, a first Pareto front approximation is computed and the second phase is initialized. To improve the current front, the standard EHI is used to select the points that contribute the most to the current hypervolume, subject to the respective constraints. A stopping criterion is devised based on the average EHI at each iteration. KEMOCO is evaluated against NSGAII using standard performance indicators, showing good performance in approximating the fronts subject to expensive constraints.

More recently, Feliot et al. (2017) put forward a comprehensive kriging-based Bayesian framework for single and multiobjective optimization with constraints. The approach is referred to as Bayesian multiobjective optimization (BMOO), and uses the EHI and PoF as infill criteria. The EHI is computed and optimized using sequential Monte Carlo simulations. The dominated hypervolume is defined using an extended domination rule, which handles objectives and constraints in a unified way. BMOO is intended to be used in problems for 3 or more objectives, as several algorithms for the exact bi-objective EHI contributions already exist. The computational cost is significantly reduced by using approximations instead of exact EHI contributions. Experimental results show that BMOO is able to find solutions efficiently on multiple benchmark problems, outperforming EMO, MEI-SPOT and EMmI (the latter is discussed later in this section). The authors mainly attribute this good performance to the fact that BMOO is designed to handle non-linear constraints, whereas the other algorithms were adapted to do so.

Henkenjohann et al. (2007) (see also Henkenjohann et al. (2005)) propose an approach, here referred to as Multi-EI, where the sequential search is guided using the preferences of the decision-maker during the optimization process, by defining *desirable* regions in the response space. They argue that with multiple responses, the scaling and the demands on quality for the responses often differ. The algorithm does not aim to approximate the entire Pareto front, but to yield the subset of Pareto points that are most valuable for the decision-maker. This is done by evaluating *desirability functions* that quantify the decision-maker's preferences for each response, such that the larger the desirability, the better the quality of the outcome for that response. The individual desirabilities are then combined into the *desirability index* (DI) of a given decision vector $\mathbf{x}_i$ as the geometric mean of the desirability function (d) values for all m responses:

$$DI[\mathbf{f}(\mathbf{x}_i)] = \prod_{j=1}^{m} [d(f_j(\mathbf{x}_i))]^{w_j} \tag{20}$$

subject to $\sum_{j=1}^{m} w_j = 1$, where w_j represents the weight (preference) of a particular response $j = 1, ..., m$. These indices are used to derive a *multivariate expected improvement*, which is used as the infill criterion. As discussed in Svenson (2011), this method is vulnerable to the choice of preferences, and not suitable for more than 3-4 objectives.

An alternative to computing the improvement in the objective space, is to consider the progress in the design space. An example of this approach appears in the *stepwise uncertainty reduction* (SUR) algorithm of Picheny (2015). The SUR criterion selects the point with the lowest uncertainty in the multiobjective PoI. The measure of uncertainty as defined in SUR is similar to the entropy measure used in Villemonteix et al. (2009) and Chevalier et al. (2014). The main advantage of the SUR algorithm is that it is scale-invariant since it does not focus on progress in terms of objective values, which can be of great advantage when dealing with objectives of different nature (Picheny, 2015). On the other hand, it is computationally expensive as it requires numerical integration embedded in the optimization loop. According to Hernández L. et al. (2016), SUR is feasible for 3 objectives at most.

Svenson & Santner (2016) proposes a multiobjective improvement function based on the *modified maximin fitness function* (referred to as EMmI), and additionally outlines a general approach for modeling multiple responses through a multivariate kriging model that allows for dependent as well as independent response functions. The proposed model assumes that the objective vector $\mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), ..., f_m(\mathbf{x})\}$ at a solution $\mathbf{x}$ is an observation from an m-variate Gaussian process $\boldsymbol{F(x)}$:

$$\boldsymbol{F(x)} = \boldsymbol{\beta} + \boldsymbol{AM(x)} \tag{21}$$

where $\boldsymbol{A} = a_{i,j}$ is a symmetric $m \times m$ positive definite matrix containing the covariances between each couple of objectives $i, j \in \{1, .., m\}$, $\boldsymbol{\beta} = (\beta_1, ..., \beta_m)^T$ and $\boldsymbol{M(x)} = [M_1(\mathbf{x}), ..., M_m(\mathbf{x})]^T$ is an $m \times 1$ vector of mutually independent stationary Gaussian processes with mean zero and unit variance. Dependencies between response functions can be captured by an A matrix having a nondiagonal form.

The infill criterion is based on the following generalization of the maximin fitness function (Balling, 2003):

$$I_{\mathcal{M}}[\mathbf{f}(\mathbf{x})] = \left[- \max_{\mathbf{x}_i \in \mathcal{P}_n^D} \min_{j=1,\ldots,m} [f_j(\mathbf{x}) - f_j(\mathbf{x}_i)] \right] \times 1_E \quad (22)$$

where $\mathcal{P}_n^D = \{\mathbf{x}_1, ..., \mathbf{x}_p\}$ is the current Pareto set for n computed responses so far. Thus, $p \leq n$ and $\mathcal{P}_n^\Theta = \{\mathbf{f}(\mathbf{x}_1), ..., \mathbf{f}(\mathbf{x}_p)\}$ are the respective response vectors. The indicator function 1_E is a binary operator which equals 1 when $-\max_{\mathbf{x}_i \in \mathcal{P}_n^D} \min_{j=1,\ldots,m}[f_j(\mathbf{x}) - f_j(\mathbf{x}_i)] < 0$, and 0 otherwise (see Bautista (2009) for more details on the non-truncated version of this function). As shown in Svenson & Santner (2016), using Eq. 22 as an infill criterion in the search for the Pareto front is essentially equivalent to using the additive binary ϵ indicator (Zitzler et al., 2003). The experimental performance of the proposed criterion is comparable to the EHI and outperforms MEI-SPOT; yet, it is clear the implementation and computation of $I_{\mathcal{M}}$ is significantly less complex and expensive than EHI, as it does not require any piecewise integrations and its implementation is just three nested loops. The results also show that the independent model in general outperforms the dependent model when there is no prior information on potential dependencies among the objectives.

Chugh et al. (2016) presents a kriging-assisted reference vector guided evolutionary algorithm (K-RVEA), which is the kriging-assisted version of the RVEA algorithm of Cheng et al. (2016). It is capable of dealing with as many as 10 objectives and 10 dimensions (as opposed to the previously discussed approaches, which are limited to 2-3 objectives). Populations are sequentially updated with points that are selected either because they have maximum kriging variance or mimimum Angle Penalized Distance (APD); the latter indicator is designed to dynamically balance the convergence (by measuring the distance between the candidate solutions), and diversity (by measuring the angle between the candidate solutions and a number of reference vectors) of the Pareto frontier (Cheng et al., 2016). Choosing infill points based on maximum variance prioritizes diversity, while minimum APD focuses more convergence. The performance of K-RVEA is compared to MOEA/D-EGO, SMS-EGO, ParEGO and its non-kriging-based version RVEA. On average, K-RVEA outperforms all the other algorithms when dimensionality is high (e.g., 10 in the experiments), for up to 10 objectives, in terms of computational time, hypervolume and inverted generational distance.

6. Kriging-based multiobjective optimization algorithms for stochastic problems

Very few articles in the literature have made an attempt at *noisy* multi-objective simulation optimization. In general, all the algorithms surveyed in this section follow a sequential procedure as depicted in Figure 2. In addition to the noisy outputs, the distribution of the finite computational budget now becomes a crucial issue, as the evaluation of candidate points normally requires multiple replications in order to achieve sufficient accuracy. For a fixed replication budget, this results in a lower number of infill points that can be sampled, which may have an important impact on the overall performance of the algorithm.

We summarize the kriging-based algorithms surveyed in Table 3. All these algorithms assume homogeneous simulation noise, meaning that the variance of the noise does not depend on $\mathbf{x}$, as opposed to heterogeneous noise (see Picheny et al. (2013) for a review and performance evaluation of kriging-based methods for single-objective problems with homogeneous noise; for problems with heterogeneous noise see Jalali et al. (2017)). We identify the following noise handling strategies among the surveyed algorithms:

1. Static resampling (SR): Replicate the objective values for each design a fixed number of times and take the average. This method reduces the variance of the objective estimate by a factor of $\sqrt{b}$, where b is the fixed number of replications, but increases the computational cost by a factor b (Jin & Branke, 2005).

2. Kriging with nugget effect (KNE): The term "nugget" refers to a variation or error in the measurement (Kleijnen, 2015). This nugget is often used to model the effect of white noise in the observations, under the assumption that the variance of the noise is homogeneous; thus this variance is a constant. The nugget effect is introduced in the kernel structure by adding a hyperparameter that models the variability in the observations; the kriging metamodel then loses its interpolating nature (see Cressie (1993), Rasmussen (2006) and Gramacy & Lee (2012) for further details).

3. Re-interpolation (RI): The RI method was introduced in Forrester et al. (2006). It first fits an initial kriging metamodel with nugget effect (i.e., a non-interpolating metamodel) to the observations, and then fits an interpolating metamodel to the predictions of the first KNE metamodel. The second kriging metamodel is then used to make predictions during optimization.

4. Rolling Tide Evolutionary Algorithm (RTEA): This algorithm was presented in Fieldsend & Everson (2015), and uses evolutionary operators to assign re-evaluations only on promising points; bad solutions are evaluated only once. The selection of promising candidates is based on their current dominance relation and the number of prior replications. An interesting feature in RTEA is that only during the first phase of the algorithm new points are sampled; the second phase is for improving the accuracy of the sampled solutions.

Horn et al. (2017) apply the SMS-EGO algorithm to noisy settings, using two naive and two advanced noise handling strategies. The first naive strategy is to ignore the effect of noise and treat the problem as deterministic. Replications are simply omitted, so more points in the design space can be sampled. The other naive strategy is static resampling. The two more advanced strategies are the one used in the RTEA algorithm of Fieldsend & Everson (2015), and a reinforced strategy, which simply treats the problem as deterministic at the beginning to

Algorithm	Reference	Infill criterion	Noise handling strategy	Search space	Numerical experiments
Noisy SMS-EGO	Horn et al. (2017)	LCB and EHI	SR, KNE and RTEA	Discretized and continuous	Problem: practical/analytical Decision variables: 5 Objectives: $2 \sim 3$ Constraints: No
Noisy SMS-EGO Noisy SExI-EGO	Koch et al. (2015)	LCB and EHI	SR and RI	Discretized	Problem: practical/analytical Decision variables: $2 \sim 8$ Objectives: 2 Constraints: No
PESMO	Hernández L. et al. (2016) Hernández L. et al. (2014)	Predictive ES	KNE	Discretized	Problem: practical/analytical Decision variables: $3 \sim 6$ Objectives: $2 \sim 4$ Constraints: No
ϵ-PAL	Zuluaga et al. (2016) Zuluaga et al. (2013)	ϵ-Pareto	KNE	Discretized	Problem: practical Decision variables: $3 \sim 11$ Objectives: 2 Constraints: No

Table 3: Summary of kriging-based algorithms for stochastic multiobjective problems.

collect a set of candidate points; it then performs extra replications on this candidate set to determine the non-dominated points with reduced variance. However, it is clear that with the latter method, due to sampling variability, superior solutions may be ignored and inferior solutions may be selected during the search.

The experimental setting consists of a few analytical test functions and a practical machine learning problem. The test functions are contaminated with homogeneous Gaussian noise, and the practical problem is known to be affected by heterogeneous noise; yet this noise is treated as homogeneous, which in turn yield bad results in performance. On average, the RTEA algorithm was able to outperform the other noise handling strategies. Moreover, the authors analyze the effect of using a nugget when fitting the metamodels and conclude that not ignoring the effect of the noise by characterizing it during optimization is fundamental to obtain reliable Pareto-optimal solutions. It is also emphasized the importance of considering heterogeneous noise in practice.

Koch et al. (2015) adapts SMS-EGO (Ponweiser et al., 2008a) and SExI-EGO (Emmerich et al., 2011) for noisy evaluations. The RI method of Forrester et al. (2006) is employed to deal with the inherent simulation noise and compared to using static resampling; thus, KNE metamodels are also used. Extensive experiments were carried out on a set of biobjective test functions with a maximum of 8 dimensions, and on two practical problems. Results show that these noisy variants of SMS-EGO and SExI-EGO perform relatively well with the RI method; RI is found to be crucial in order to obtain reliable results. However, the performance on the practical problems was significantly worse due to the higher noise levels. The authors emphasize that ignoring the noise level during the optimization process results in considerably worse approximations; the requirement of replicating on the same point significantly reduces the number of optimal solutions sampled, and thus the overall performance of the algorithms.

The Predictive Entropy Search for Multiobjective Optimization (PESMO) algorithm is developed in Hernández L. et al. (2016). PESMO selects as infill point the one that is expected to yield the largest decrease in the entropy of the predictions that belong to the current Pareto front. This approach is referred to as *predictive entropy search* (Hernández L. et al., 2014). To handle the noise, KNE metamodels are fitted to the different responses, and instead of resampling the objectives through the expensive simulator, samples are taken from these KNE metamodels. This technique is widely used in single-objective Bayesian optimization (see e.g., Frazier et al. (2009)). As the reduction in entropy is formulated as a sum across the objectives, PESMO allows for the evaluation of new design points on subsets of objectives, instead of requiring a value for all the responses in each iteration. This results in a computational cost that is linear in the number of objectives, and thus is relatively cheap. Another advantage is that, analogous to SUR (discussed in Section 5.2), PESMO measures the progress in the design space (i.e., the Pareto set), as opposed to measure it in the objective space with standard quality indicators that rely on noisy observations.

The authors compared the performance of PESMO against ParEGO, SMS-EGO, SExI-EGO, SUR, and an expensive non-kriging-assisted version of itself. The performance metric used is based on the relative difference between the hypervolume of the Pareto front of the actual objectives and the Pareto front obtained by the algorithm. Results show that PESMO outperforms all other algorithms significantly, for both the noisy and noiseless cases. For the biobjective case, SExI-EGO performs worst in average, followed by ParEGO, SMS-EGO and SUR (the performance of SUR, though, is significantly worse in the noisy case). However, ParEGO is at least 3 times faster than PESMO, and 56 times faster than SUR on average. Results with a 4-objective function show that PESMO yields nearly 35% better quality Pareto fronts than ParEGO, and 20% better than SMS-EGO. The superior performance of PESMO is attributed to its ability to identify the most noisy areas in the response surface of the objectives, in order to evaluate those observations with extra replications.

Zuluaga et al. (2016) propose the ϵ-Pareto Active Learning algorithm (ϵ-PAL), an adaptive learning technique, regulated by the parameter ϵ, to predict a set of Pareto optimal solutions that cover the true Pareto front with ϵ tolerance. The algorithm is an extension of the PAL algorithm of Zuluaga et al. (2013), and predicts an ϵ-accurate Pareto set by training multiple KNE metamodels with subsets of points in the decision space. The kriging predictions of each point $\mathbf{x}$ are used to maintain an uncertainty region around the objective values associated with $\mathbf{x}$, allowing to make statistical inferences about the Pareto-optimality of every point in the decision space. ϵ-PAL selects as infill point the one with the highest uncertainty region around it, as these are the points that require more replications.

The experimental results show that ϵ-PAL outperforms PAL based on the percentage of the true Pareto set found by the algorithm, and requires shorter runtimes. In addition, ϵ-PAL returns an ϵ-accurate Pareto front instead of a dense approximation of it. It is often the case that small differences in performance are not significant to the decision-maker, and thus not worth the substantial extra computational effort to determine the true best. This is conveyed with the parameter ϵ in the proposed algorithm, analogous to defining a so-called *indifference zone*, a well-known procedure in *ranking and selection* (Boesel et al., 2003). In general, ϵ-PAL also outperforms ParEGO both in terms of function evaluations required (being 30-70% lower than with ParEGO), and in terms of computation times (reduced by a factor of up to 420).

7. Conclusion

In this article, we surveyed the most relevant kriging-based MO algorithms for deterministic and stochastic problems, in the context of numerically expensive simulators. It is clear that kriging-based algorithms for deterministic problems are at a more advanced stage: here, important progress has been made in developing multiobjective infill criteria, and algorithms that exploit such criteria. Yet, most of these criteria remain very expensive to calculate, limiting the suitability of the algorithms to problems with at most 2-4 objectives. An exception is the K-RVEA algorithm, which has been shown to outperform other algorithms both in terms of computational time and quality of the Pareto front obtained for problems with up to 10 objectives.

The development of kriging-based MO algorithms for stochastic problems is still in its infancy. The main issue is how to handle the noise; only two very recent algorithms (PESMO by Hernández L. et al. (2016) and ϵ-PAL by Zuluaga et al. (2016)) take the noise into account in the kriging model itself and replicate only on competitive designs, both showing promising results. Yet, their approach implicitly assumes that the noise is homogeneous. Strikingly, none of the algorithms so far incorporates a kriging approach that can deal with heterogeneous noise. The powerful *stochastic kriging* approach, developed by Ankenman et al. (2010), the *variational heteroscedastic gaussian process regression* developed by Lázaro-Gredilla & Titsias (2011), or the *kriging with modified nugget effect* by Yin et al. (2011) can be used in this case. The use of any of these methods during optimization remains a major opportunity for future research.

Surprisingly none of the algorithms surveyed for stochastic problems use probabilistic dominance or a MORS procedure in order to asses the dominance relationship between the points and/or allocate computational budget proportional to the noise affecting the outputs. While PESMO and ϵ-PAL make a first effort to distribute budget based on noise, a substantial amount of work remains to be done in this regard. In addition, given the scarce research on the topic, the further development of MORS procedures could provide an important step forward (see Section 3.2).

Finally, another important challenge for the multiobjective community in general is the modeling of the preferences of the decision-maker (see e.g., Branke et al. (2017) and Pedro & Takahashi (2013)). Finding an entire approximation of the Pareto front is not always in the interest of the decision-maker. Instead, some areas of the objective space (e.g., so-called "knees" (Branke et al., 2004)) might be more interesting. Computational budget should be allocated to search solutions on areas of the Pareto front that are interesting to the decision-maker, especially when the evaluation of solutions is expensive and we need to rely on surrogate approximations. Kriging metamodels can be exploited to model the decision-maker preferences, as discussed in the Multi-EI algorithm (Henkenjohann et al., 2007), and more recently proposed in Hakanen & Knowles (2017) using the ParEGO algorithm, but extensive further work can be done in this direction.

Acknowledgment

This research was supported by the Research Foundation-Flanders, grant number G076815. The authors would like to thank the anonymous reviewers for their comments and suggestions, as these improved the paper significantly.

References

Abraham, A., Jain, L. C., & Goldberg, R. (2005). *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

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

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, *3*, 397–422.

Auger, A., Bader, J., Brockhoff, D., & Zitzler, E. (2012). Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, *425*, 75–103.

Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, *66*, 55–69.

Bader, J., & Zitzler, E. (2011). Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.*, *19*, 45–76.

Balling, R. (2003). The maximin fitness function; multi-objective city and regional planning. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 1–15). Springer.

Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.

Basseur, M., & Zitzler, E. (2006). A preliminary study on handling uncertainty in indicator-based multiobjective optimization. In *Applications of Evolutionary Computing* (pp. 727–739). Springer Berlin Heidelberg.

Bautista, D. C. T. (2009). *A sequential design for approximating the pareto front using the expected pareto improvement function*. Ph.D. thesis The Ohio State University.

Boesel, J., Nelson, B. L., & Kim, S.-H. (2003). Using ranking and selection to "clean up" after simulation optimization. *Operations Research*, *51*, 814–825.

Bonnel, H., & Collonge, J. (2014). Stochastic optimization over a pareto set associated with a stochastic multi-objective optimization problem. *Journal of Optimization Theory and Applications*, *162*, 405–427.

Bonnel, H., & Collonge, J. (2015). Optimization over the pareto outcome set associated with a convex bi-objective optimization problem: theoretical results, deterministic algorithm and application to the stochastic case. *Journal of Global Optimization*, *62*, 481–505.

Box, G. E., Draper, N. R. et al. (1987). *Empirical model-building and response surfaces* volume 424. Wiley New York.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Branke, J., Corrente, S., Greco, S., & Gutjahr, W. (2017). Efficient pairwise preference elicitation allowing for indifference. *Computers & Operations Research*, *88*, 175 – 186.

Branke, J., Deb, K., Dierolf, H., & Osswald, M. (2004). Finding knees in multi-objective optimization. In *International conference on parallel problem solving from nature* (pp. 722–731). Springer.

Branke, J., Zhang, W., & Tao, Y. (2016). Multio-bjective ranking and selection based on hypervolume. In *Winter Simulation Conference (WSC), 2016* (pp. 859–870). IEEE.

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, .

Broomhead, D. S., & Lowe, D. (1988). *Radial basis functions, multi-variable functional interpolation and adaptive networks*. Technical Report Royal Signals and Radar Establishment Malvern (United Kingdom).

Cheng, R., Jin, Y., Olhofer, M., & Sendhoff, B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *20*, 773–791.

Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., & Richet, Y. (2014). Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, *56*, 455–465.

Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., & Sindhya, K. (2016). A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, .

Chugh, T., Sindhya, K., Hakanen, J., & Miettinen, K. (2017). A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing*, (pp. 1–30).

Coello, C. A. C., & Sierra, M. R. (2004). A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Mexican International Conference on Artificial Intelligence* (pp. 688–697). Springer.

Couckuyt, I., Deschrijver, D., & Dhaene, T. (2012). Towards efficient multi-objective optimization: multiobjective statistical criterions. In *Evolutionary Computation (CEC), 2012 IEEE Congress on* (pp. 1–8). IEEE.

Couckuyt, I., Deschrijver, D., & Dhaene, T. (2014). Fast calculation of multi-objective probability of improvement and expected improvement criteria for pareto optimization. *Journal of Global Optimization*, *60*, 575–594.

Cressie, N. (1993). *Statistics for spatial data*. New York: John Wiley & Sons.

Das, I., & Dennis, J. E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, *14*, 63–69.

Davins-Valldaura, J., Moussaoui, S., Pita-Gil, G., & Plestan, F. (2017). ParEGO extensions for multi-objective optimization of expensive evaluation functions. *Journal of Global Optimization*, *67*, 79–96.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*, 182–197.

Dellino, G., Lino, P., Meloni, C., & Rizzo, A. (2007). Enhanced evolutionary algorithms for multidisciplinary design optimization: A control engineering perspective. In A. Abraham, C. Grosan, & H. Ishibuchi (Eds.), *Hybrid Evolutionary Algorithms* (pp. 39–76). Berlin, Heidelberg: Springer.

Dellino, G., Lino, P., Meloni, C., & Rizzo, A. (2009). Kriging metamodel management in the design optimization of a cng injection system. *Mathematics and Computers in Simulation*, *79*, 2345–2360.

Dellino, G., Lino, P., Meloni, C., Rizzo, A., Bonomo, C., Fortuna, L., Giannon, P., & Graziani, S. (2012). Simulation-optimisation in modelling ionic polymer-metal composites actuators. *International Journal of Modelling, Identification and Control*, *17*, 8–18.

Diaz-Manriquez, A., Toscano, G., Barron-Zambrano, J. H., & Tello-Leal, E. (2016). A review of surrogate assisted multiobjective evolutionary algorithms. *Intell. Neuroscience*, *2016*, 19–.

Ebden, M. (2015). Gaussian Processes: A Quick Introduction. *ArXiv e-prints*, . arXiv:1505.02965.

Emmerich, M. T., & Deutz, A. H. (2018). A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural computing*, *17*, 585–609.

Emmerich, M. T., Deutz, A. H., & Klinkenberg, J. W. (2011). Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *Evolutionary Computation (CEC), 2011 IEEE Congress on* (pp. 2147–2154). IEEE.

Emmerich, M. T., Giannakoglou, K. C., & Naujoks, B. (2006). Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, *10*, 421–439.

Feldman, G., Hunter, S. R., & Pasupathy, R. (2015). Multi-objective simulation optimization on finite sets: optimal allocation via scalarization. In *Proceedings of the 2015 Winter Simulation Conference* (pp. 3610–3621). IEEE Press.

Feliot, P., Bect, J., & Vazquez, E. (2017). A bayesian approach to constrained single-and multi-objective optimization. *Journal of Global Optimization*, *67*, 97–133.

Fieldsend, J. E., & Everson, R. M. (2005). Multi-objective optimisation in the presence of uncertainty. In *2005 IEEE Congress on Evolutionary Computation* (pp. 243–250). volume 1.

Fieldsend, J. E., & Everson, R. M. (2015). The rolling tide evolutionary algorithm: A multiobjective optimizer for noisy optimization problems. *IEEE Transactions on Evolutionary Computation*, *19*, 103–117.

da Fonseca, V. G., Fonseca, C. M., & Hall, A. O. (2001). Inferential performance assessment of stochastic optimisers and the attainment function. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 213–225). Springer.

Forrester, A., Sobester, A., & Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. Chichester: John Wiley & Sons.

Forrester, A. I., & Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, *45*, 50–79.

Forrester, A. I., Keane, A. J., & Bressloff, N. W. (2006). Design and analysis of "noisy" computer experiments. *AIAA journal*, *44*, 2331–2339.

Frazier, P., Powell, W., & Dayanik, S. (2009). The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, *21*, 599–613.

Frazier, P. I. (2018). A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, .

Fricker, T. E., Oakley, J. E., & Urban, N. M. (2013). Multivariate gaussian process emulators with nonseparable covariance structures. *Technometrics*, *55*, 47–56.

Fu, M. C. (2014). Handbook of simulation optimization.

Giagkiozis, I., & Fleming, P. J. (2015). Methods for multi-objective optimization: An analysis. *Information Sciences*, *293*, 338–350.

Gong, D., na Qin, N., & yan Sun, X. (2010). Evolutionary algorithms for multi-objective optimization problems with interval parameters. In *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)* (pp. 411–420).

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

Gutjahr, W. J., & Pichler, A. (2016). Stochastic multi-objective optimization: a survey on non-scalarizing methods. *Annals of Operations Research*, *236*, 475–499.

Hakanen, J., & Knowles, J. D. (2017). On using decision maker preferences with parego. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 282–297). Springer.

Hansen, M. P., & Jaszkiewicz, A. (1998). Evaluating the quality of approximations to the non-dominated set, .

Henkenjohann, N., Göbel, R., Kleiner, M., & Kunert, J. (2005). An adaptive sequential procedure for efficient optimization of the sheet metal spinning process. *Quality and Reliability Engineering International*, *21*, 439–455. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.732`. doi:`10.1002/qre.732`.

Henkenjohann, N., Kunert, J., & a, a. (2007). An efficient sequential optimization approach based on the multivariate expected improvement criterion. *Quality Engineering*, *19*, 267–280.

Hennig, P., & Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, *13*, 1809–1837.

Hernández L., D., Hernandez-Lobato, J., Shah, A., & Adams, R. (2016). Predictive entropy search for multi-objective bayesian optimization. In *International Conference on Machine Learning* (pp. 1492–1501).

Hernández L., J. M., Hoffman, M. W., & Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems* (pp. 918–926).

Hoffman, M. D., Brochu, E., & de Freitas, N. (2011). *Portfolio Allocation for Bayesian Optimization.*. Technical Report.

Horn, D., Dagge, M., Sun, X., & Bischl, B. (2017). First investigations on noisy model-based multi-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 298–313). Springer.

Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, *10*, 477–506.

Hunter, S. R., Applegate, E. A., Arora, V., Chong, B., Cooper, K., Rincón-Guevara, O., & Vivas-Valencia, C. (2019). An introduction to multiobjective simulation optimization. *ACM Trans. Model. Comput. Simul.*, *29*, 7:1–7:36. URL: `http://doi.acm.org/10.1145/3299872`. doi:`10.1145/3299872`.

Hupkens, I., Emmerich, M., & Deutz, A. (2014). Faster computation of expected hypervolume improvement. *arXiv preprint arXiv:1408.7114*, .

Jalali, H., Van Nieuwenhuyse, I., & Picheny, V. (2017). Comparison of kriging-based algorithms for simulation optimization with heterogeneous noise. *European Journal of Operational Research*, .

Jeong, S., & Obayashi, S. (2005). Efficient global optimization (ego) for multi-objective problem and data mining. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on* (pp. 2138–2145). IEEE volume 3.

Jin, Y., & Branke, J. (2005). Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on evolutionary computation*, *9*, 303–317.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, *21*, 345–383.

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

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

Kleijnen, J. P., & Mehdad, E. (2014). Multivariate versus univariate kriging metamodels for multi-response simulation models. *European Journal of Operational Research*, *236*, 573–582.

Kleijnen, J. P. C. (2015). *Design and Analysis of Simulation Experiments*. (2nd ed.). NY: Springer.

Knowles, J. (2006). ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans. Evol. Comput.*, *10*, 50–66.

Knowles, J. D., Corne, D., & Reynolds, A. P. (2009). Noisy multiobjective optimization on a budget of 250 evaluations. In *EMO* (pp. 36–50). Springer.

Koch, P., Wagner, T., Emmerich, M. T., Bäck, T., & Konen, W. (2015). Efficient multi-criteria optimization on noisy machine learning problems. *Applied Soft Computing*, *29*, 357–370.

Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, *52*, 119–139.

Law, A. M. (2015). *Simulation Modeling and Analysis*. McGraw-Hill, New York.

Lázaro-Gredilla, M., & Titsias, M. K. (2011). Variational heteroscedastic gaussian process regression. In *Proceedings of the 28th International Conference on International Conference on Machine Learning* ICML'11 (pp. 841–848). USA.

Lee, L. H., Chew, E. P., Teng, S., & Goldsman, D. (2010). Finding the non-dominated pareto set for multi-objective simulation models. *IIE Transactions*, *42*, 656–674.

Lemieux, C. (2009). *Monte carlo and quasi-monte carlo sampling*. Springer Science & Business Media.

Li, G., Li, M., Azarm, S., Al Hashimi, S., Al Ameri, T., & Al Qasas, N. (2009). Improving multi-objective genetic algorithms with adaptive design of experiments and online metamodeling. *Structural and Multidisciplinary Optimization*, *37*, 447–461.

Li, H., Lee, L. H., Chew, E. P., & Lendermann, P. (2015). MO-COMPASS: a fast convergent search algorithm for multi-objective discrete optimization via simulation. *IIE Transactions*, *47*, 1153–1169.

Li, M., Li, G., & Azarm, S. (2008). A kriging metamodel assisted multi-objective genetic algorithm for design optimization. *Journal of Mechanical Design*, *130*, 031401.

Liu, W., Zhang, Q., Tsang, E., Liu, C., & Virginas, B. (2007). On the performance of metamodel assisted moea/d. In *International Symposium on Intelligence Computation and Applications* (pp. 547–557). Springer.

MacKay, D. J. (1998). Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, *168*, 133–166.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, *26*, 369–395.

Martinez F., J., & Herrero P., D. (2016). Kriging-based infill sampling criterion for constraint handling in multi-objective optimization. *Journal of Global Optimization*, *64*, 97–115. URL: https://doi.org/10.1007/s10898-015-0370-8. doi:10.1007/s10898-015-0370-8.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, *58*, 1246–1266.

Meckesheimer, M., Booker, A. J., Barton, R. R., & Simpson, T. W. (2002). Computationally inexpensive metamodel assessment strategies. *AIAA journal*, *40*, 2053–2060.

Miettinen, K. (1999). Nonlinear multiobjective optimization, volume 12 of international series in operations research and management science.

Miettinen, K., & Mäkelä, M. M. (2002). On scalarizing functions in multiobjective optimization. *OR spectrum*, *24*, 193–213.

Mockus, J. (2012). *Bayesian approach to global optimization: theory and applications* volume 37. Springer Science & Business Media.

Parr, J. M., Forrester, A. I., Keane, A. J., & Holden, C. M. (2012). Enhancing infill sampling criteria for surrogate-based constrained optimization. *Journal of Computational Methods in Sciences and Engineering*, *12*, 25–45.

Pedro, L. R., & Takahashi, R. H. (2013). Decision-maker preference modeling in interactive multiobjective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 811–824). Springer.

Picheny, V. (2015). Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, *25*, 1265–1280.

Picheny, V., Wagner, T., & Ginsbourger, D. (2013). A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, *48*, 607–626.

Ponweiser, W., Wagner, T., Biermann, D., & Vincze, M. (2008a). Multiobjective optimization on a limited budget of evaluations using model-assisted\mathcal {S}-metric selection. In *International Conference on Parallel Problem Solving from Nature* (pp. 784–794). Springer.

Ponweiser, W., Wagner, T., & Vincze, M. (2008b). Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on* (pp. 3515–3522). IEEE.

Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. Citeseer.

Riquelme, N., Von Lücken, C., & Baran, B. (2015). Performance metrics in multi-objective optimization. In *Latin American Computing Conference (CLEI), 2015* (pp. 1–11). IEEE.

Rubinstein, R. Y., & Kroese, D. P. (2016). *Simulation and the Monte Carlo method*. John Wiley & Sons.

Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical science*, (pp. 409–423).

Santner, T. J., Williams, B. J., & Notz, W. I. (2013). *The design and analysis of computer experiments*. Springer Science & Business Media.

Shimoyama, K., Sato, K., Jeong, S., & Obayashi, S. (2013). Updating kriging surrogate models based on the hypervolume indicator in multi-objective optimization. *Journal of Mechanical Design*, *135*, 094503.

Singh, P., Couckuyt, I., Ferranti, F., & Dhaene, T. (2014). A constrained multi-objective surrogate-based optimization algorithm. In *2014 IEEE Congress on Evolutionary Computation (CEC)* (pp. 3080–3087).

Svenson, J., & Santner, T. (2016). Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Computational Statistics & Data Analysis*, *94*, 250–264.

Svenson, J. D. (2011). *Computer experiments: Multiobjective optimization and sensitivity analysis*. Ph.D. thesis The Ohio State University.

Syberfeldt, A., Ng, A., John, R. I., & Moore, P. (2010). Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling. *European Journal of Operational Research*, *204*, 533–544.

Tabatabaei, M., Hakanen, J., Hartikainen, M., Miettinen, K., & Sindhya, K. (2015). A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. *Structural and Multidisciplinary Optimization*, *52*, 1–25.

Trautmann, H., Mehnen, J., & Naujoks, B. (2009). Pareto-dominance in noisy environments. In *Proceedings of the Eleventh Conference on Congress on Evolutionary Computation* CEC'09 (pp. 3119–3126). Piscataway, NJ, USA: IEEE Press.

Ulmer, H., Streichert, F., & Zell, A. (2003). Evolution strategies assisted by gaussian processes with improved preselection criterion. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on* (pp. 692–699). IEEE volume 1.

Van Beers, W. C., & Kleijnen, J. P. C. (2003). Kriging for interpolation in random simulation. *Journal of the Operational Research Society*, *54*, 255–262.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Villemonteix, J., Vazquez, E., & Walter, E. (2009). An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, *44*, 509–534.

Voß, T., Trautmann, H., & Igel, C. (2010). New uncertainty handling strategies in multi-objective evolutionary optimization. In *Parallel Problem Solving* *from Nature, PPSN XI* (pp. 260–269). Berlin, Heidelberg: Springer Berlin Heidelberg.

Voutchkov, I., & Keane, A. (2010). Multi-objective optimization using surrogates. In *Computational Intelligence in Optimization* (pp. 155–175). Springer.

Wagner, T., Emmerich, M., Deutz, A., & Ponweiser, W. (2010). On expected-improvement criteria for model-based multi-objective optimization. *Parallel Problem Solving from Nature, PPSN XI*, (pp. 718–727).

Wang, G. G., & Shan, S. (2007). Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical design*, *129*, 370–380.

Yin, J., Ng, S., & Ng, K. (2011). Kriging metamodel with modified nugget-effect: The heteroscedastic variance case. *Computers & Industrial Engineering*, *61*, 760–777.

Zaefferer, M., Bartz-Beielstein, T., Naujoks, B., Wagner, T., & Emmerich, M. (2013). A case study on multi-criteria optimization of an event detection software under limited budgets. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 756–770). Springer.

Zhan, D., Cheng, Y., & Liu, J. (2017). Expected improvement matrix-based infill criteria for expensive multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, .

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*, 712–731.

Zhang, Q., Liu, W., Tsang, E., & Virginas, B. (2010). Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Transactions on Evolutionary Computation*, *14*, 456–474.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, *1*, 32 – 49.

Zitzler, E., Brockhoff, D., & Thiele, L. (2007). The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary multi-criterion optimization* (pp. 862–876). Springer.

Zitzler, E., Knowles, J., & Thiele, L. (2008). Quality assessment of pareto set approximations. *Multiobjective Optimization*, (pp. 373–404).

Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature* (pp. 832–842). Springer.

Zitzler, E., Laumanns, M., Thiele, L., Fonseca, C. M., & da Fonseca, V. G. (2002). Why quality assessment of multiobjective optimizers is difficult. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation* (pp. 666–674). Morgan Kaufmann Publishers Inc.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*, 117–132.

Zuluaga, M., Krause, A., & Püschel, M. (2016). ε-pal: an active learning approach to the multi-objective optimization problem. *The Journal of Machine Learning Research*, *17*, 3619–3650.

Zuluaga, M., Sergent, G., Krause, A., & Püschel, M. (2013). Active learning for multi-objective optimization. In *International Conference on Machine Learning* (pp. 462–470).