Bi-objective ranking and selection using stochastic kriging

Peer-reviewed author version

ROJAS GONZALEZ, Sebastian; Branke, Juergen & VAN NIEUWENHUYSE, Inneke
(2024) Bi-objective ranking and selection using stochastic kriging. In: European journal of operational research,.

DOI: 10.1016/j.ejor.2024.11.008
Handle: http://hdl.handle.net/1942/45150

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Decision Support

Bi-objective ranking and selection using stochastic kriging

Sebastian Rojas Gonzalez [a,c,*], Juergen Branke [b], Inneke Van Nieuwenhuyse [a,c]

[a] Department of Information Technology, Ghent University, Belgium
[b] Department of Operations, Warwick University, United Kingdom
[c] FlandersMake@UHasselt and Data Science Institute, Hasselt University, Belgium

ARTICLE INFO

Keywords:
Multiple criteria analysis
Multiobjective simulation optimization
Stochastic kriging
Multiobjective ranking and selection

ABSTRACT

We consider bi-objective ranking and selection problems, where the goal is to correctly identify the Pareto-optimal solutions among a finite set of candidates for which the objective function values have to be estimated from noisy evaluations. When identifying these solutions, the noise perturbing the observed performance may lead to two types of errors: solutions that are truly Pareto-optimal may appear to be dominated, and solutions that are truly dominated may appear to be Pareto-optimal. We propose a novel Bayesian bi-objective ranking and selection method that sequentially allocates extra samples to competitive solutions, in view of reducing the misclassification errors when identifying the solutions with the best expected performance. The approach uses stochastic kriging to build reliable predictive distributions of the objectives, and exploits this information to decide how to resample. The experiments are designed to evaluate the algorithm on several artificial and practical test problems. The proposed approach is observed to consistently outperform its competitors (a well-known state-of-the-art algorithm and the standard equal allocation method), which may also benefit from the use of stochastic kriging information.

1. Introduction

In *multi-objective* or *multi-criteria* optimization problems, the goal is to find the set of solutions that reveal the essential trade-offs between the objectives (i.e., where no single objective can be improved without negatively affecting any other objective); these solutions are referred to as *non-dominated or Pareto-optimal*. When shown in the objective space, they form the *Pareto front*, and the set of the corresponding decision vectors is referred to as the *Pareto set*. Depending on the type of problem, the Pareto front may have different geometries (e.g. concave, convex, linear, disconnected). In real-life problems, the location and geometry of the Pareto front are evidently unknown, and a discrete set of solutions is often used to approximate it (Hunter et al., 2019; Miettinen, 1999).

This article considers multi-objective problems where the objectives can only be evaluated by means of stochastic experiments (e.g., stochastic simulation). Consequently, the performances are distorted by noise. Such problems occur frequently in real life. For instance, in manufacturing and distribution problems (e.g., lot sizing in manufacturing plants), it is common to maximize the expected profit of the production system, while also minimizing profit risk; in transportation problems (e.g., constructing optimal routes for dial-a-ride services) it is common to minimize tardiness of deliveries while also using minimal resources; and in the healthcare sector it is often important to minimize both the waiting time of patients and the idle time of doctors.

In this paper, we assume that the analyst has a finite set of solutions at hand, for which they have already observed the objective outcomes for a given number of replications. Such a set may result, for instance, from running a multi-objective metaheuristic (see, e.g., Fieldsend and Everson 2015), or a multi-objective simulation optimization procedure (see e.g., Rojas-Gonzalez and Van Nieuwenhuyse 2020). The goal of *multi-objective ranking and selection* (MORS) is to correctly identify the true Pareto-optimal solutions among such a finite set, by adding extra replications to solutions in the most informative way.

In the current literature on multi-objective stochastic simulation optimization (see e.g., Feliot et al. 2017, Hernández-Lobato et al. 2016, Horn et al. 2017), the ranking and selection phase is often neglected: the sample means of the available objective evaluations are then directly used to determine the Pareto-optimal set, disregarding the sampling variability. Evidently, this may lead to two types of errors: designs that are truly Pareto-optimal might appear to be dominated, or designs that are truly dominated might appear to be Pareto-optimal. Chen and Lee (2010) refer to these errors as *Error Type 1* and *Error Type 2* respectively, whereas (Hunter et al., 2019) refer to them as *misclassification by exclusion* (MCE) and *misclassification by*

* Corresponding author at: Department of Information Technology, Ghent University, Belgium.
E-mail address: sebastian.rojasgonzalez@ugent.be (S.R. Gonzalez).

https://doi.org/10.1016/j.ejor.2024.11.008
Received 10 October 2022; Accepted 5 November 2024

inclusion (MCI). Throughout this paper, we use the MCE/MCI terminology. The accuracy of the objective estimates can obviously be improved by allocating extra samples, yet the resampling budget is naturally limited. Allocating this budget among the candidates in the finite set is not trivial. *Static resampling* (allocating the budget uniformly across all alternatives) tends to be inefficient, in particular in settings with limited budget and complex noise structures, as it often wastes a lot of replications on inferior solutions (Rojas-Gonzalez et al., 2020).

Literature on the MORS problem is relatively recent, and still very limited (Hunter et al., 2019). In this paper, we propose a sequential MORS method that uses *stochastic kriging* (SK) metamodels (Ankenman et al., 2010) to build predictive distributions of the objectives, based on the sample means and variances after replication. This allows the proposed method to exploit correlations in the objective outcomes across different solutions, even in the presence of heteroscedastic noise. Under a few mild conditions, and for few input dimensions, SK metamodels provide reliable predictions that are superior to using sampling information directly (Staum, 2009), as the noise in the observed performance is considered in the predictions. We exploit the SK information in a novel heuristic to determine which solutions to resample, and show how other MORS algorithms can also benefit from using such metamodel information. In addition, we propose two screening procedures to reduce the computational burden at each iteration. While some preliminary ideas of the method presented here were published in Rojas-Gonzalez et al. (2019), all the allocation decisions we put forward in this paper are novel and show competitive performance against the state-of-the-art.

The remainder of this article is organized as follows: Section 2 gives an overview of the relevant literature. Section 3 defines the problem and Section 4 gives the necessary stochastic kriging background. Section 5 explains the proposed sampling criteria and outlines the proposed algorithm, referred to as SK-MORS. We design the experiments to evaluate the performance of the algorithm in Section 6, analyze the results in Section 7 and conclude in Section 8.

2. Related work

The ranking and selection problem has been widely studied in the single-objective case, and has been approached in different ways. One of the most common goals is to maximize the *probability of correct selection* (PCS), where under some mild conditions, convergence to the solution with the true best expected performance can be theoretically guaranteed (see e.g., Boesel et al. 2003, Frazier 2014, Kim and Nelson 2006). In some cases, however, an extremely large replication budget might be required to differentiate two solutions with almost identical performance, while such small differences in performance are likely not relevant to the decision maker. Therefore, *indifference zone* (IZ) approaches aim at guaranteeing the selection of the best solution *or* a solution within a given user-defined quantity from the best (known as the *probability of good selection*). This user-specified quantity defines the indifference zone, and represents the smallest difference worth detecting (see e.g., Boesel et al. 2003, Fan et al. 2016, Kim and Nelson 2001).

Relative to the single-objective case, the literature on multi-objective ranking and selection is scarce; a good overview can be found in Hunter et al. (2019). The multi-objective PCS (hereafter denoted mPCS) is defined as the probability of correctly identifying the entire Pareto set, and only this set (Branke & Zhang, 2019). For both the single and multi-objective case, the true PCS must be estimated, usually via Monte Carlo simulation, which often leads to a computational bottleneck (Chen & Lee, 2010). The multi-objective case is evidently more complex, as the true Pareto front is often continuous and unknown (and thus approximated with a relatively large discrete set), and the dominance relationships between solutions are extremely sensitive to simulation replications (see e.g. the results in Rojas-Gonzalez et al. 2020).

One of the most widely used MORS methods is the *Multi-objective Optimal Computing Budget Allocation* (MOCBA), proposed in Lee et al. (2010), built upon the well-known single-objective ranking and selection approach OCBA (Chen & Lee, 2010). As discussed by the authors, the MORS problem can be formulated in several ways (e.g., maximizing the mPCS or minimizing the classification errors when identifying the non-dominated set). The proposed allocation rules use numerical approximations to estimate which solutions are likely to be dominated. In Chen and Lee (2010) a simplified version of the original MOCBA is proposed, which avoids most of these issues. Another relevant framework is the *SCORE allocations for bi-objective ranking and selection* (Applegate et al., 2020; Feldman & Hunter, 2018; Feldman et al., 2015), which aims to allocate replications to maximize the rate of decay of the probability of wrongly identifying the true Pareto set, and is asymptotically optimal.

More recently, Andradóttir and Lee (2021) proposed a multi-objective IZ procedure to estimate the Pareto set with the true best expected performance with statistical guarantees. The work focuses on providing a stopping criterion for replicating designs until a predetermined desired mPCS is achieved, which is desirable in some MORS settings, but at the expense of a large replication budget due to the approximation of several parameters. Another related work appears in Binois et al. (2015), where by means of kriging metamodels, predictive distributions are built over the objectives and conditional simulations are used to estimate the probability that any given point in the objective space is dominated; the method is found to be very sensitive to the (unknown) geometry of the Pareto front, and is computationally expensive. To the best of our knowledge, there are only two other multi-objective IZ procedures in the literature: Branke and Zhang (2019) and Teng et al. (2010). The latter work clearly highlights the shortcomings of the first, but remains limited to the biobjective case, and is computationally demanding.

A different approach appears in the M-MOBA and M-MOBA-HV algorithms (Branke & Zhang, 2019; Branke et al., 2016), which apply a Bayesian approach to determine the solution that, when replicated further, is expected to yield the maximum *information value* (Chick et al., 2010). In the M-MOBA algorithm, this solution is the one with the highest probability of changing the current Pareto set, whereas in M-MOBA-HV, it is the one leading to the largest change in the observed *hypervolume* (a widely used performance metric in deterministic multi-objective optimization, that quantifies the volume of the objective space dominated by a given set of points; see Auger et al. 2012, Zitzler et al. 2007 for further details on hypervolume).

Another related stream of research comes from the *evolutionary multi-objective optimization* community, which should in principle apply a MORS procedure to assess the fitness of the solutions in each iteration, if the observations are noisy. This issue is currently neglected, however, and the sample mean is commonly used as the fitness estimate, regardless of the sampling variability. Several authors have looked at defining the concept of dominance in noisy multi-objective optimization (see e.g., Basseur and Zitzler 2006, Fieldsend and Everson 2005, Hughes 2001, Teich 2001), but there is no consensus in the literature on the formalization of this concept. In Trautmann et al. (2009) and Voß et al. (2010), *probabilistic dominance* is defined by taking the volume of the confidence intervals on the objective outcomes, and looking at the center point of these volumes in order to determine the dominance relationship. A relatively simple yet well performing approach is to allocate additional samples to solutions that survive from one generation to the next in the evolutionary phase of the algorithm, as is proposed in the Rolling Tide Evolutionary Algorithm (Fieldsend & Everson, 2015). Another related approach is *dynamic resampling*, which varies the additional number of samples based on the estimated variance of the observed objective values. It aims to assess the observed responses at a particular confidence level before determining dominance, and to avoid unnecessary resampling (see e.g., Di Pietro et al. 2004, Syberfeldt et al. 2010).

3. Problem definition

A multi-objective optimization problem can be defined as follows:

$$\min[f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})].$$

In this expression, $\mathcal{X} \subset \mathbb{R}^d$; the notation $\mathbf{x} = [x_1, \ldots, x_d]$ then refers to a *decision vector* (also referred to as *design*, or simply as *point*) of dimension d. The vector-valued function $f : \mathcal{X} \rightarrow \mathbb{R}^m$ links each decision vector with its objective values $(f_1, \ldots, f_m)$, where m denotes the number of objectives. In such a multi-objective setting, the concept of *dominance* is used to compare solutions:

Definition 1. For two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\mathcal{X}$, $\mathbf{x}_1 \prec \mathbf{x}_2$ means $\mathbf{x}_1$ dominates $\mathbf{x}_2$ iff $f_j(\mathbf{x}_1) \leq f_j(\mathbf{x}_2), \forall j \in \{1, \ldots, m\}$, and $\exists j \in \{1, \ldots, m\}$ such that $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$.

The solution to the multi-objective problem consists of the set of $\mathbf{x}$ vectors that are not dominated by any other solution. This set of decision vectors is referred to as the *Pareto set*; often, we can only achieve a discrete approximation of this set, consisting of a finite number of decision vectors. The corresponding set of objective vectors is called the *Pareto front*.

In the stochastic case, the estimate for a given objective at a given decision vector $\mathbf{x}_i$ is usually calculated as the sample mean of that objective over r_i independently and identically distributed (IID) replications:

$$\bar{f}_j(\mathbf{x}_i) = \sum_{k=1}^{r_i} \bar{f}_j^k(\mathbf{x}_i)/r_i, \forall i = 1, \ldots, n; j = 1, \ldots, m \tag{1}$$

where $\bar{f}_j^k(\mathbf{x}_i)$ denotes the simulated performance of $\mathbf{x}_i$ on objective j at the kth replication. As these replication outcomes are noisy measurements of the objective performance $f_j(\mathbf{x}_i)$, the resulting sample mean is also noisy:

$$\bar{f}_j(\mathbf{x}_i) = f_j(\mathbf{x}_i) + \epsilon_j(\mathbf{x}_i) \tag{2}$$

The noise $\epsilon_j(\mathbf{x}_i)$ is commonly assumed to be independent among the different objectives; in practice, it tends to be *heteroscedastic* (Ankenman et al., 2010; Kim & Nelson, 2006), meaning that it varies throughout the search space.

In this paper, we focus on the bi-objective case (i.e., $m = 2$ objectives); to simplify notation, we sometimes use $\bar{f}_{ij}$ to denote $\bar{f}_j(\mathbf{x}_i)$. The posterior distribution for the unknown true function value of objective j at point $\mathbf{x}_i$ is

$$\{f_{ij}\} \sim \mathcal{N}(\bar{f}_{ij}, \frac{\bar{s}_{ij}^2}{r_i}), \forall i = 1, \ldots, n; j = 1, 2 \tag{3}$$

where $\bar{f}_{ij}$ and $\bar{s}_{ij}^2$ denote the sample mean and sample variance of objective j at design i, based on the r_i replications. Evidently, Eq. (3) expresses the Bayesian belief that each objective value is normally distributed, with the posterior variance decreasing as r_i increases. Adding r_i' additional simulation replications, yielding an average performance of $\bar{f}_{ij}'$, updates the sample mean as follows (Branke & Zhang, 2019):

$$\bar{f}_{ij} \leftarrow \frac{r_i \bar{f}_{ij} + r_i' \bar{f}_{ij}'}{r_i + r_i'}. \tag{4}$$

Consequently, the dominance relationships among the alternatives might change, thereby changing the estimated Pareto front (and, correspondingly, the estimated Pareto set).

The goal of this research is to develop an efficient and effective algorithm to allocate an additional (but limited) number of replications among a finite number of competing designs, in view of minimizing the expected total number of misclassification errors at the end of the algorithm. We thus focus on solving the following optimization problem (see also Lee et al. 2010):

$$\min_{r_1', \ldots, r_n'} E(\text{MCE+MCI}) \tag{5}$$

$$\text{s.t.} \sum_{i=1}^{n} r_i' \leq R$$

$$r_i' \geq 0, i = 1, \ldots, n$$

where r_i' denotes the number of additional replications on design i, R is the replication budget available after performing r_i replications on each design i, and E(MCE+MCI) is the expected total number of misclassification errors. As the exact number of errors cannot be computed because the true Pareto optimal points are in general unknown, this optimization problem is analytically intractable. In this paper, we propose a sequential heuristic that aims to solve the problem using *stochastic kriging* (Ankenman et al., 2010).

4. Stochastic kriging

Assume that we have observed the performance of an arbitrary objective function $f(\mathbf{x}_i)$ on a finite set of n designs of dimension d, using r_i IID replications per design. *Stochastic kriging* (SK) represents the simulation output in each replication k by the following model:

$$\bar{f}^k(\mathbf{x}_i) = \beta + M(\mathbf{x}_i) + \epsilon^k(\mathbf{x}_i), \tag{6}$$

where β is a constant, and $M(\mathbf{x}_i)$ is a realization of a mean 0 random field. The term $\epsilon^k(\mathbf{x}_i)$ denotes the *intrinsic* noise in stochastic simulation, typically assumed to be naturally independent and identically distributed across replications, with $E[\epsilon^k(\mathbf{x}_i)] = 0$ and $Var[\epsilon^k(\mathbf{x}_i)] = \tau^2(\mathbf{x}_i)$. As $\tau^2(\mathbf{x}_i)$ depends on the location of $\mathbf{x}_i$, the model naturally allows for *heteroscedastic* noise. The random field M is assumed to exhibit spatial correlation, meaning that the outcomes $M(\mathbf{x}_i)$ and $M(\mathbf{x}_h)$ will tend to be similar when $\mathbf{x}_i$ is close to $\mathbf{x}_h$ in the design space. Ankenman et al. (2010) refer to the random nature of M as *extrinsic* uncertainty, as it is imposed on the problem in view of developing the model. In the design and analysis of computer experiments, the standard assumption for M is that it is a Gaussian random field.

The core of the extrinsic spatial correlation model is the *covariance function* or *kernel*, which is used to quantify the covariance between the outcomes of any two decision vectors $\mathbf{x}_i$ and $\mathbf{x}_h$. In this work we use the (stationary) squared exponential kernel, also known as Gaussian kernel (Eq. (7)), and the Matérn kernel (Eq. (8)):

$$cov[M(\mathbf{x}_i), M(\mathbf{x}_h)] = v^2 \exp\left[-\sum_{q=1}^{d}\left(\frac{|x_{i,q} - x_{h,q}|}{l_q}\right)^2\right] \tag{7}$$

$$cov_{\nu=3/2}[M(\mathbf{x}_i), M(\mathbf{x}_h)] = v^2\left[1 + \sqrt{3}\sum_{q=1}^{d}\frac{|x_{i,q} - x_{h,q}|}{l_q}\right] \times \exp\left[-\sum_{q=1}^{d}\frac{|x_{i,q} - x_{h,q}|}{l_q}\right] \tag{8}$$

where v^2 and l_q $(q = 1, \ldots, d)$ are *hyperparameters* that denote the process variance and the length-scale of the process along dimension q, respectively. These hyperparameters, along with the constant β in Eq. (6), are estimated from the available data (i.e., the n design vectors and their corresponding sample means), usually by means of *maximum likelihood estimation* (MLE). For further details on other popular kernels, see Chapter 5 in Rasmussen and Williams (2005).

Using the obtained MLE estimates ($\hat{\beta}$, $\hat{v}^2$ and $\hat{l}_q$) as plug-in estimators, the analyst can then estimate the *belief* of the SK model with respect to the true function outcome, at each design point $\mathbf{x}_i$:

$$\{f_i\} \sim \mathcal{N}(\hat{f}(\mathbf{x}_i), \hat{s}^2(\mathbf{x}_i)), \forall i = 1, \ldots, n \tag{9}$$

where $\hat{f}(\mathbf{x}_i)$ and $\hat{s}^2(\mathbf{x}_i)$ denote the *stochastic kriging predictor* and the *predictor uncertainty* at design point $\mathbf{x}_i$, respectively. These are given by:

$$\begin{aligned}\hat{f}(\mathbf{x}_i) &= \hat{\beta} + \hat{\Sigma}_M(\mathbf{x}_i, \cdot)^T[\hat{\Sigma}_M + \Sigma_\epsilon]^{-1}(\bar{\mathbf{f}} - \hat{\beta}\mathbf{1}_n) \\ \hat{s}^2(\mathbf{x}_i) &= \hat{\Sigma}_M(\mathbf{x}_i, \mathbf{x}_i) - \hat{\Sigma}_M(\mathbf{x}_i, \cdot)^T[\hat{\Sigma}_M + \Sigma_\epsilon]^{-1}\hat{\Sigma}_M(\mathbf{x}_i, \cdot)\end{aligned} \tag{10}$$

$$+\frac{\gamma^T\gamma}{\mathbf{1}_n^T[\hat{\Sigma}_M+\Sigma_\epsilon]^{-1}\mathbf{1}_n} \tag{11}$$

with $\gamma = 1-\mathbf{1}_n^T[\hat{\Sigma}_M+\Sigma_\epsilon]^{-1}\hat{\Sigma}_M(\mathbf{x}_i,\cdot)$.

In Eq. (10) (also referred to as the *posterior mean*) and Eq. (11) (also referred to as the *MSE* or *posterior variance*), $\bar{\mathbf{f}} = [\bar{f}(\mathbf{x}_1),\dots,\bar{f}(\mathbf{x}_n)]^T$ is the $n\times 1$ vector containing the sample means, and $\mathbf{1}_n$ is a $n\times 1$ vector of ones. The $n\times n$ matrix $\hat{\Sigma}_M$ contains the kernel result for each pair of design vectors (i.e., $cov[M(\mathbf{x}_h), M(\mathbf{x}_{h'})]$, for $h,h' = 1,\dots,n$), using the plug-in estimators. Analogously, the $n\times 1$ vector $\hat{\Sigma}_M(\mathbf{x}_i,\cdot)$ contains the kernel results for the covariance between the function outcome at the design vector $\mathbf{x}_i$ for which the predictor is calculated, and each of the n design vectors in the set (i.e., $cov[M(\mathbf{x}_i), M(\mathbf{x}_h)]$, for $h = 1,\dots,n$). The $n\times n$ matrix Σ_ϵ contains the covariances between the sample means of the n design vectors implied by the intrinsic noise: $\text{diag}[\bar{s}^2(\mathbf{x}_1)/r_1,\dots,\bar{s}^2(\mathbf{x}_n)/r_n]$.

If each of the points in the set were replicated an infinite number of times, Σ_ϵ would reduce to a zero matrix, implying that the estimated sample means would be noise-free. In that case, the stochastic kriging estimates in Eqs. (10) and (11) would reduce to the well-known *ordinary kriging* expressions (Kleijnen, 2015): the predictor would perfectly coincide with the sample mean, and the predictor uncertainty at each point would reduce to zero. Yet, in real life, the replication budget is finite. Consequently, in all practical settings, the SK predictor in Eq. (10) will *not* coincide with the sample mean at the observed points; yet, the difference between them *tends to zero* as the noise on the sample means is reduced. We exploit this crucial property in our proposed MORS procedure, which, as described in the next section, uses distinct stochastic kriging models for each objective.

5. Proposed MORS procedure

Let S be the set containing the n design vectors; at each location $\mathbf{x}_i$ ($i = 1,\dots,n$), we have performed r_i IID replications for each objective j. Fitting a SK model to this data allows us to determine the predictor (Eq. (10)) and MSE (Eq. (11)) for each point in S and for each objective j. We thus dispose of two types of information for each design vector: sample information (sample mean and variance) and SK information (predictor and MSE). While we expect SK to yield more accurate performance estimates than the sample means, these predictions may have a high MSE associated to them. Thus, we are interested in sampling on designs where *both* estimates are very different, as we expect these to get closer to each other (and closer to the true performance) as we increase the number of replications.

Consequently, we can identify one estimated Pareto front (and corresponding Pareto set) based on the sample means, and another one based on the SK information. Both fronts (and both sets) will be equivalent *in the limit* (i.e., when the observational noise has been reduced to zero). Hereafter, the notation $\overline{\{\cdot\}}$ refers to a result determined using sample information, while the notation $\widehat{\{\cdot\}}$ denotes a result based on SK model information. For example, $\bar{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_i$ denote the objective vectors for design $\mathbf{x}_i$ identified using sample information and predicted information respectively, while $\widehat{PS}$ and $\widehat{PF}$ denote the set of Pareto-optimal points identified based on the model predictions. In what follows, we first discuss the sampling criteria used in the proposed procedure, followed by the outline of the resulting algorithm.

5.1. Proposed sampling criteria

Our aim is to allocate a finite budget of extra replications to the points in S in such a way that we reduce the expected number of MCE/MCI errors. Ideally, in the end, the Pareto-optimal points within the finite set should be identified correctly. To do this we propose two sampling criteria that the algorithm uses to rank and select the point(s) to which extra replications will be added: the *posterior distance (PD)*, and the *expected hypervolume difference (EHVD)*. These two criteria serve different goals: while the PD focuses on improving the prediction accuracy, EHVD allocates budget to those points for which errors in the current sample mean estimates, caused by the noise, have the biggest impact on the identification of the front. In the proposed algorithm, we select the point(s) by maximizing both criteria *simultaneously*, in each iteration. This is explained in detail in Section 5.2, when we discuss the algorithm outline.

5.1.1. Posterior distance

As the SK predictor accounts for the intrinsic noise on the sample mean, this causes the predictor value to differ from the sample mean, and the MSE to differ from zero. Obviously, if the sample means *were* noise-free, they would reflect the true function values, and the SK predictor would reduce to the *ordinary kriging* predictor (as *ordinary kriging* is an exact interpolator). For this reason, we propose to quantify the attractiveness of adding more samples to a given location $\mathbf{x}_i$ by means of that point's *posterior distance* (PD):

$$\text{PD}_i = \sqrt{\sum_{j=1}^{m}\left[\,|\,\bar{f}_j(\mathbf{x}_i)-\hat{f}_j(\mathbf{x}_i)\,|+\hat{s}_{ij}\right]^2}, \qquad \forall i\in S. \tag{12}$$

Adding replications to points with a high PD value reduces the noise on the sample mean, such that the difference between $\bar{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_i$ reduces. Again, with infinite sampling budget, both would coincide with the vector of true function values, so applying this sampling criterion would *in the limit* reduce the sum of expected MCE and MCI errors to zero (see also Applegate et al. 2020). Note that Eq. (12) resembles the Euclidean distance between both vectors. While in Bayesian optimization adding the MSE to the prediction value aids in exploring areas of the search space where the uncertainty is high (Cox & John, 1992; Jones, 2001), here it is added under the root to inflate the PD at points with high posterior variance.

5.1.2. Expected hypervolume difference

In reality, budgets are not infinite. As the MCE and MCI errors result from a misclassification of both the true Pareto-optimal points and the truly dominated points, we would like the MORS algorithm to focus its resampling efforts on those points where the current value of the (noisy) sample mean may impact the correct identification of the front. Following the rationale of the SK metamodel, the SK predictors can be used as estimates, as they reflect the expected value of the models' *belief* of the true function outcome, based on the model assumptions and the currently available information. To quantify *the extent* to which a given point's sample mean may impact the correct identification of the front, we propose to look at the *expected hypervolume difference* for that point, if its sample mean were replaced by its SK predictor. By definition, the hypervolume associated with an arbitrary front PF is given by

$$HV(PF,\mathbf{r}) = \Lambda\left(\bigcup_{z\in PF}\{z' : z\prec z'\prec\mathbf{r}\}\right). \tag{13}$$

where $\mathbf{r}$ denotes the reference point, and the notation Λ refers to the Lebesgue measure. The *expected hypervolume difference* for a given point $\mathbf{x}_i$ is then determined as:

$$\text{EHVD}_i = |\,HV(\overline{PF},\mathbf{r}) - HV\left(\Delta_{PF},\mathbf{r}\right)\,|\,. \tag{14}$$

with $\Delta_{PF} = \overline{PF}\backslash\bar{\mathbf{f}}_i\cup\hat{\mathbf{f}}_i$. If replacing $\bar{\mathbf{f}}_i$ by $\hat{\mathbf{f}}_i$ does not yield *any* change in the estimated front, the result is zero. This can only happen when *both* vectors $\bar{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_i$ are located in the dominated space, as shown in Fig. 1(a), or when both vectors are exactly the same (which is very unlikely). In all other cases in the figure, EHVD_i results in a strictly positive value (illustrated by the shaded area), as the front changes. Fig. 1(b) shows a case where *both* estimates are on the front and mutually non-dominated; Fig. 1(c) and (d) show the cases when both estimates are on the front, but $\hat{\mathbf{f}}\prec\bar{\mathbf{f}}$ or vice versa; and Fig. 1(e) and (f) illustrate the cases where one of them turns out to be dominated by some other vector $\bar{\mathbf{g}}$ close to the front (e.g., $\hat{\mathbf{f}}\prec\bar{\mathbf{g}}\prec\bar{\mathbf{f}}$).

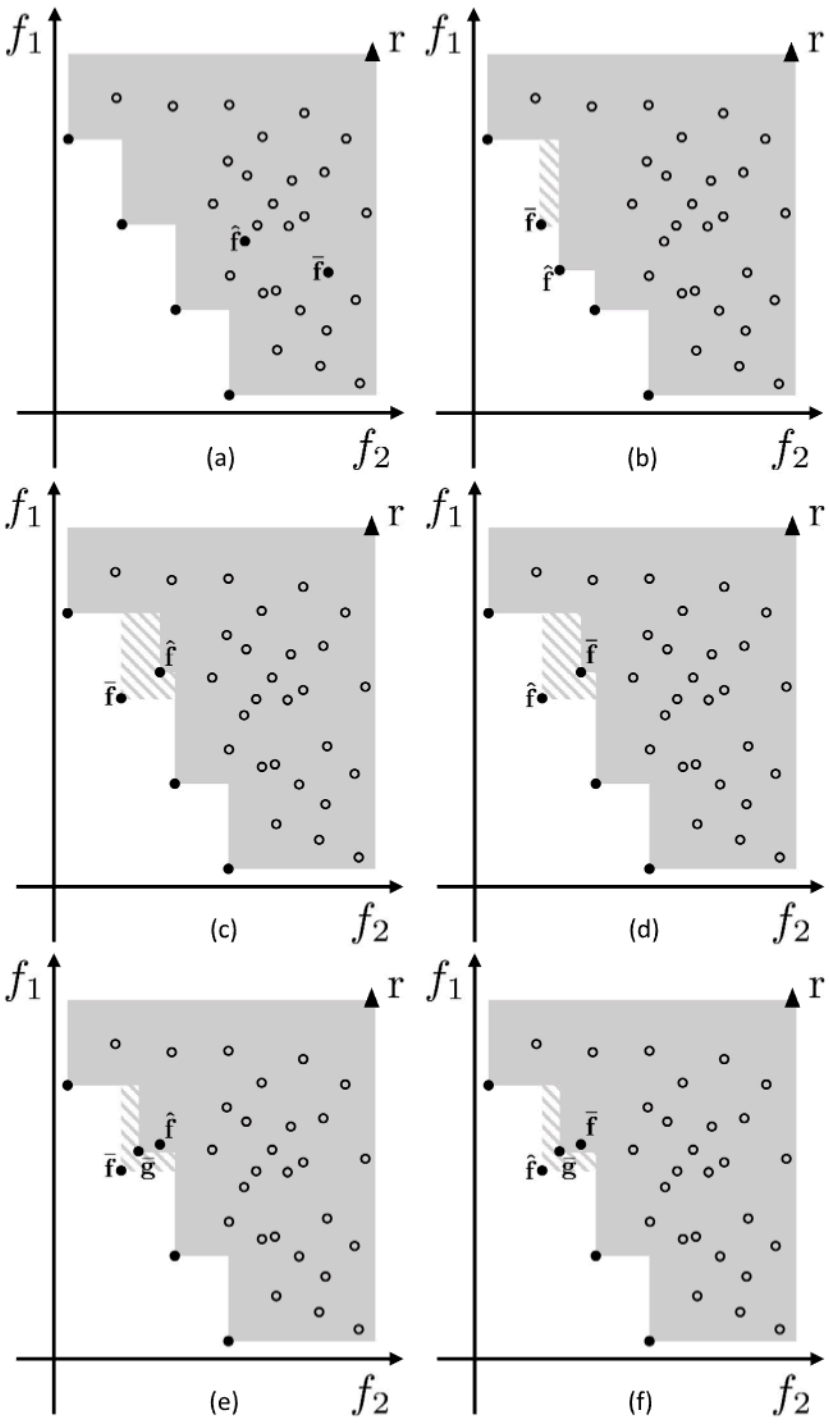


Fig. 1. The different cases to be considered when calculating the EHVD.

5.2. Algorithm outline

The procedure is outlined in Algorithm 1. At the initial stage, we assume $b_0 > 1$ replications have been already allocated to each point in S, information the algorithm uses to fit a SK metamodel to each objective. The metamodels are then used to predict each response and respective prediction errors on these sampled points. The current dominated and non-dominated sets are computed based on both the sample and predicted means. For ease of reference, we summarize the different sets of points used in Table 1.

The Screening Box procedure outlined in the algorithm is intended to screen out points that are unlikely to change the front when they receive extra replications (we refer to these as *inferior points*). In stochastic single-objective simulation–optimization, such procedures are widely used prior to running the ranking and selection procedure (see e.g., Boesel et al. 2003). In the stochastic multi-objective case, it is not straightforward to determine which points are statistically superior to others, as some points might be better on one or more objectives with a certain probability, while simultaneously being inferior on one or more other objectives with a different probability (Chen & Lee, 2010). The procedure uses the lower and upper confidence bounds of the objectives at the different points; as both the model predictions and the sample means are normally distributed, we can estimate these bounds as $\phi_j(\mathbf{x}_i) \pm \omega\sqrt{\sigma_j^2(\mathbf{x}_i)}, j = 1, \ldots, m$, where ϕ and σ are, respectively, the

Table 1
Overview of the different sets of points used in the proposed algorithm.

Notation	Description
S	Finite set of sampled points.
$\widetilde{S}$	Set of sampled points that have not been screened out.
$\overline{PS}, \overline{PF}$	Pareto set and front based on sample information.
$\widehat{PS}, \widehat{PF}$	Pareto set and front based on predicted information.
$\overline{UCB}, \overline{LCB}$	Set of upper and lower confidence bounds of all points in S using sample means.
$\widehat{UCB}, \widehat{LCB}$	Set of upper and lower confidence bounds of all points in S using predicted means.

performance estimate and associated variance for design $\mathbf{x}_i$, and ω is the critical value chosen by the analyst (Cox & John, 1992; Jones, 2001). In the experiments, we use 99%, unless explicitly stated otherwise.

For each objective, the *maximum upper confidence bound* among all the *non-dominated* points is determined (denoted by $\bar{u}_j$ for the sample means, and $\hat{u}_j$ for the predicted means in lines 15 – 16 of Algorithm 1). Note that, because we are minimizing the objectives, the combination of these j upper confidence bounds defines a (hyper)box in the objective space dominated by the *current* Pareto front, both for the sample means and for the predicted means. By contrast, the procedure considers the combination of all *lower confidence bounds* for each *dominated point*. If the result falls outside the hyperbox, the corresponding design is considered to be inferior, and is screened out from the *current* iteration. As the set of inferior points is recalculated in each iteration, it might be considered in further iterations.

While we acknowledge the heuristic nature of the procedure, the experiments show that it helps to reduce the number alternatives (which reduces computational effort, most importantly for the EHVD calculations) without affecting the overall performance of the entire allocation procedure. Consequently, with Screening Box, it is highly unlikely to screen out points that are truly non-dominated. Appendix A discusses an alternative procedure (Screening Band), which is more aggressive in screening out points compared to the Screening Box procedure. Yet, as evidenced by the results in this appendix, it shows a higher risk of excluding truly non-dominated points. Consequently, it may not be suitable for settings with high or strongly heterogeneous noise.

The algorithm then calculates the EHVD and PD values for the remaining points, in view of selecting the point(s) that will receive extra replications; those points that have already reached the maximum number of replications b_{max} are not considered. Based on the results, a Pareto front can be distinguished (denoted by $\widetilde{PF}$), for which both criteria are simultaneously maximized (see the illustration in Fig. 2). The corresponding Pareto set $\widetilde{PS}$ is then determined, and the $\mathbf{x}_i$ locations in this set are sorted according to increasing value of r_i. Next, the algorithm iterates over the design vectors in this set, allocating one extra replication per point, until the budget B per iteration is depleted. Note that the number of points in $\widetilde{PS}$ varies per iteration, as $\widetilde{PF}$ differs per iteration.

Selecting the parameters b_0, B and b_{max} will depend on the specific total budget available, and on the number of points in the given problem. As discussed in Chen and Lee (2010), a value of $b_0 \geq 5$ is recommended to approximate the sample means and variances, and b_{max} is used to prevent the algorithms from allocating an extreme number of replications to a given point in order to ensure that the limited sampling effort is well distributed among the diverse (but competitive) solutions in the sampled set. B is chosen to balance the computational effort (small B requires many updates to the SK model) and the speed of learning from new data (large B means we make many sample allocation decisions without using the incoming information to update our SK model).

When the stopping criterion is met, the algorithm returns the *estimated* Pareto set based on the *predicted means*. We prefer $\widehat{PS}$ above $\overline{PS}$, as the SK information reflects the model's belief about the *true* function values, given all information available at the end of the algorithm. The proposed allocation procedure is thus sequential: the SK models are retrained with the new (more informative) sample means and variances at the start of each new iteration.

Algorithm 1 SK-MORS algorithm

Input:
S ▷ Set of candidate points
b_{max} ▷ Maximum replication budget per point
B ▷ Replication budget to be allocated per iteration
Output:
$\widehat{PS}$ ▷ The predicted Pareto set

$S_{max} = \emptyset$ ▷ Initialize set of points that reached b_{max}
$r_i = b_0, \forall i$ ▷ Initialize replication matrix
$B_{prev} = 0$ ▷ Initialize number of unused replications
while stopping criterion not met **do**
  $\hat{f}_j(\mathbf{x}_i) \leftarrow$ Fit a SK metamodel to each objective j
  Compute $\overline{PF}$,$\overline{PS}$,$\overline{UCB}$, $\overline{LCB}$, $\widehat{PF}$,$\widehat{PS}$,$\widehat{UCB}$ and $\widehat{LCB}$
  procedure Screening Box($\overline{UCB}$, $\overline{LCB}$, $\widehat{UCB}$, $\widehat{LCB}$)
    $I = \emptyset$ ▷ Initialize set of clearly inferior points
    $\bar{u}_j = \max_{i \in \overline{PF}} \overline{UCB}_i, \forall j$
    $\hat{u}_j = \max_{i \in \widehat{PF}} \widehat{UCB}_i, \forall j$
    $NP = S \setminus \overline{PS} \cap S \setminus \widehat{PS}$.
    for $i \in NP$ **do**
      for $j = 1 : m$ **do**
        if $\overline{LCB}_{ij} > \bar{u}_j$ **and** $\widehat{LCB}_{ij} > \hat{u}_j$ **then**
          $I \cup \{\mathbf{x}_i\}$, **break**
        end if
      end for
    end for
    return I
  end procedure
  $\widetilde{S} = S \setminus I$ ▷ Temporarily screen out inferior points
  $\widetilde{S} = \widetilde{S} \setminus S_{max}$ ▷ Remove points that have reached b_{max}
  Compute EHVD$_i$ and PD$_i$, $\forall i \in \widetilde{S}$ ▷ Eq. (14) and (12)
  $\widetilde{PF}, \widetilde{PS} \leftarrow$ Pareto front and set of the maximization of PD and EHVD
  $\widetilde{PS}_{sort} \leftarrow \widetilde{PS}$ sorted in ascending order of r_i values
  $B_{iter} = B + B_{prev}$ ▷ Use previously unused budget
  while $\widetilde{PS}_{sort}$ is not empty **or** $B_{iter} > 0$ **do**
    for $i \in \widetilde{PS}_{sort}$ **do**
      if $r_i < b_{max}$ **then**
        Allocate 1 additional replication to $\mathbf{x}_i$: $r_i = r_i + 1$
        $B_{iter} \leftarrow B_{iter} - 1$
      else
        $S_{max} \cup \{\mathbf{x}_i\}$
      end if
    end for
    $\widetilde{PS}_{sort} = \widetilde{PS}_{sort} \setminus S_{max}$
  end while
  $B_{prev} = B_{iter}$ ▷ Save unused budget
end while
Return $\widehat{PS}$

6. Test problems

In Section 6.1, we present the experimental details for a set of well-known analytical bi-objective test functions. These functions have

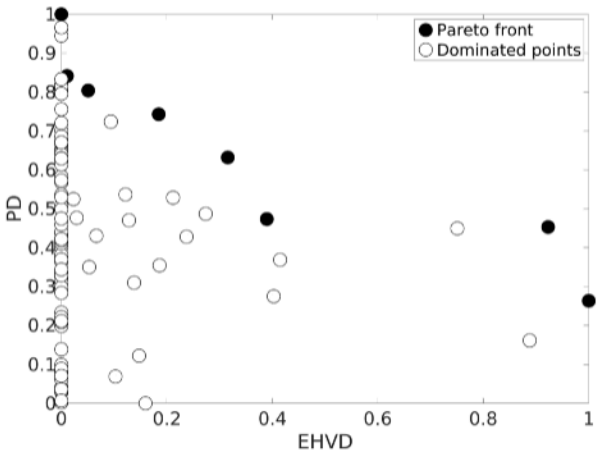


Fig. 2. Pareto front resulting from simultaneously maximizing EHVD and PD (normalized in the $[0, 1]$ interval), at an arbitrary iteration.

different Pareto front geometries, and were implemented with varying number of decision variables, and different levels of noise. They are used to evaluate the performance of the proposed allocation criteria against the standard EQUAL allocation and the MOCBA algorithm (Chen & Lee, 2010), which we consider state-of-the-art (in Appendix B we provide the allocation ratios, but the full procedure is as described in Chen and Lee 2010). Furthermore, we examine the impact of using the SK information instead of the sample means and variances in EQUAL and MOCBA as well. In Section 6.2 we evaluate the performance of our proposed algorithm against EQUAL and MOCBA on two examples motivated by real-life supply chain problems.

All algorithms are stopped when a predetermined number of additional replications R have been allocated. This is determined by the number of *iterations* performed by the algorithms and the replication budget per iteration B as $R = B \times iterations$. This is a natural criterion, as in most practical settings, the sampling budget will determine the length of the run. We compare their performances by means of the *accuracy percentage of the set (APS)* metric:

$$APS = 1 - \frac{MCE + MCI}{|S|} \tag{15}$$

where MCI and MCE are the number of misclassification errors by inclusion and exclusion, respectively, and $|S|$ is the total number of designs. The metric is an intuitive measure to evaluate the accuracy of the classification of the solutions in the sampled set, as it implicitly considers both errors to be equally important. Evidently, it takes a value between 0 and 1: if the algorithm correctly identifies the entire true non-dominated and dominated sets, there is no misclassification of points, and $APS = 1$.

6.1. Artificial test problems

We use standard test problems from the *deterministic* multi-objective literature to assess the performance of the algorithms (e.g., the ZDT, DTLZ and WFG test suites). These benchmark functions allow the analyst to construct problems with different number of objectives and decision variables; for WFG, features such as modality and separability can also be customized, using a set of shape and transformation functions. While we include results for a total of ten problems in the supplementary material (see Appendix D), here we focus on four scenarios that are representative from all the experiments we carried out. These scenarios use three test functions with different characteristics: WFG3 is a non-separable and unimodal problem with a linear Pareto front, WFG4 is a separable and multimodal problem with a concave Pareto front, and DTLZ7 is a multi-modal problem with a disconnected front (see Huband et al. 2006 for the analytical expressions and detailed characteristics of these problems).

Table 2
Summary of the experimental scenarios for the analytical test functions.

	WFG3	WFG4	DTLZ7	DTLZ7
Number of objectives m	2	2	2	2
Number of decision variables d	5	5	2	10
Number of points in S ($\|S\|$)	100	100	100	184
Size of the true Pareto set $\|PS_t\|$	20	20	50	75
Noise level	High	Medium	Low	High
Total number of *iterations*	15	30	30	15
Kernel used in SK metamodels	Squared exponential: Eq. (7)			
Low noise std. dev.	$0.001RF_j \leq \tau_j(\mathbf{x}) \leq 0.5RF_j$			
Medium noise std. dev.	$0.01RF_j \leq \tau_j(\mathbf{x}) \leq 1RF_j$			
High noise std. dev.	$1RF_j \leq \tau_j(\mathbf{x}) \leq 2RF_j$			

The four scenarios are bi-objective minimization problems with varying number of decision variables and levels of noise. The set of points S being considered is discrete and contains a fixed and known number of truly Pareto optimal points, denoted PS_t. Note that both the non-dominated *and* dominated sets of points need to be artificially generated; to do this, we ran a standard evolutionary multi-objective optimization algorithm (Deb et al., 2002) on the noise-free problem, and extracted both sets of points ensuring that the dominated set is close enough to the non-dominated set, such that it is difficult for the algorithms to converge to zero errors. Table 2 summarizes the experimental scenarios, and Fig. 3 illustrates the true objective values of the finite set of points for these test functions.

These true objective outcomes are perturbed with heterogeneous Gaussian noise. Hence, we obtain noisy observations $\bar{f}_j^k(\mathbf{x}_i) = f_j(\mathbf{x}_i) + \epsilon_j^k(\mathbf{x}_i)$, with $\epsilon_j^k(\mathbf{x}_i) \sim \mathcal{N}(0, \tau_j(\mathbf{x}_i))$ for $j = \{1, 2\}$ objectives at the kth replication. In accordance with the previous literature (Jalali et al., 2017; Picheny et al., 2013; Rojas-Gonzalez et al., 2020), we set the standard deviation of the noise $\tau_j(\mathbf{x})$, such that it varies linearly with respect to the objective values. The maximum and minimum values of $\tau_j(\mathbf{x})$ for a given objective j are linked to the range of the true values of that objective in the region of interest (i.e., $RF_j = \max_{\mathbf{x}\in S} f_j(\mathbf{x}) - \min_{\mathbf{x}\in S} f_j(\mathbf{x})$). For each individual objective j, we use the minimum noise variance at the location of that given objective's minimum (i.e., the point that would be optimal in case of single-objective minimization). Because of the trade-off between the function values in the Pareto-optimal points, this assumption automatically leads to a trade-off in the noise variance of the functions at these points: in the extremes of the bi-objective front, this variance is minimal for one objective, while higher for the other. In that way, the linear structure ensures that the resulting ranking and selection problems are non-trivial, as *none* of the Pareto-optimal points has accurate sample means on *both* objectives. Evidently, the same could have been achieved with other noise structures. As shown in Table 2, we consider three levels of noise perturbing the responses (low, medium and high), varying between $0.001RF_j$ and $2RF_j$.

6.2. Practical applications

In this section we describe two practical bi-objective problems. While both problems aim at optimizing the same two objectives, the key difference is that Problem #1 is a demand allocation problem that has 15 decision variables, and Problem #2 is an (s, S) inventory control problem with 2 decision variables. Thus, our goal with these experiments is not only to examine the convergence performance of the algorithms, but also the performance of the SK metamodels in higher dimensions. Also note that for the practical problems we use the Matérn kernel (Eq. (8)), as it is preferred over the squared exponential kernel (Eq. (7)) for practical applications (Snoek et al., 2012). For each of these scenarios, the number of designs to be considered (i.e., $|S|$) is determined after performing a search of solutions using the algorithm in Fieldsend and Everson (2015).

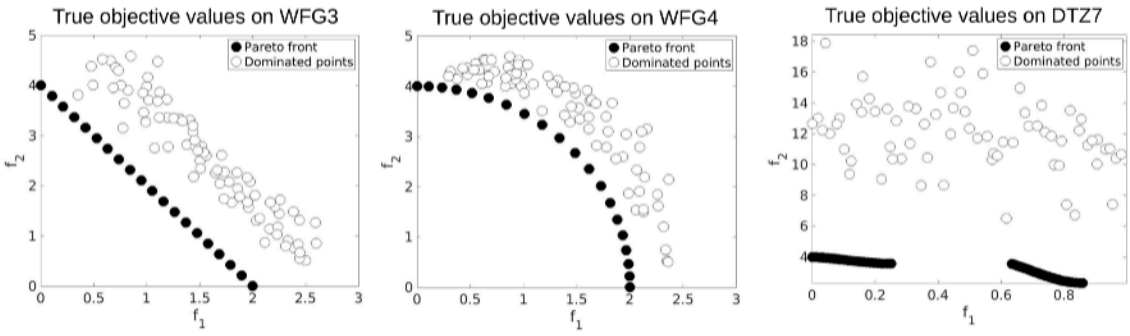


Fig. 3. True objective values for the points considered in the finite sets, and corresponding true Pareto front (PF_t, full circles) and dominated points (empty circles) for the bi-objective analytical test functions.

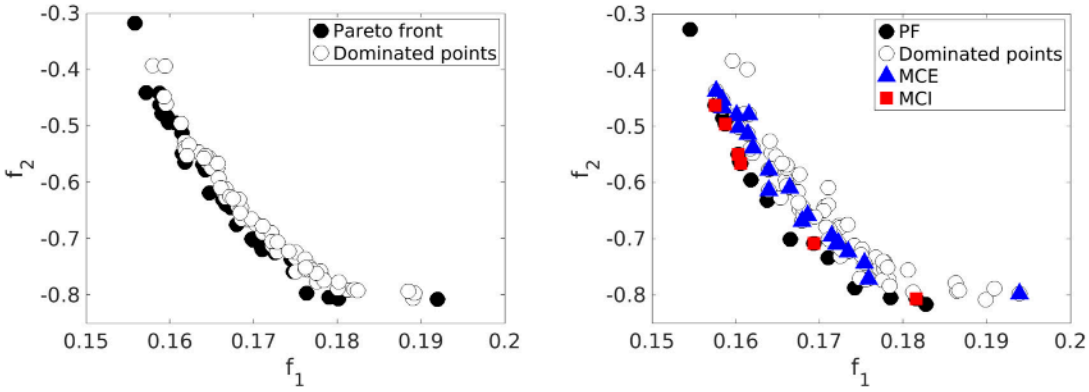


Fig. 4. Illustration of the Pareto front for the simultaneous minimization of cost (denoted f_1) and negative service level (denoted f_2).

6.2.1. Demand allocation problem

This problem is inspired by a real-life case from a company in the chemical processing industry. The company has a variety of processing units across different plant sites, each of them having unique processing capabilities with respect to product mix. Each week, the company needs to decide on the amount of product to be produced by each processing unit, and the amount of product to be shipped to each sales region, in order to meet the forecasted (weekly) demand for all product types. This problem setting has already been used as a real-life case in other articles (e.g., Jung et al. 2004), but then as a single-objective problem minimizing the total expected costs incurred (consisting of production, transportation, inventory holding and outsourcing costs) subject to achieving a minimum service level from own production. Further details of the problem formulation (based on Jung et al. (2004)) can be found in Appendix C.

In our experiment, the aim is to simultaneously minimize the expected total cost, and maximize this service level. Fig. 4 shows the trade-offs between the objectives observed for an arbitrary instance of the problem (Scenario 2). The left panel shows the approximation of the true performance, obtained by running 10^3 replications on each of the design vectors for this instance; the right panel shows the misclassification errors observed at an arbitrary iteration of the SK-MORS algorithm.

In our scenarios (see Table 3), we look at a setting with 8 sales regions, 3 production facilities and 5 product types. Every time an order is placed, the nearest facility to that sales region that can fulfill the order is selected. If the order cannot be fulfilled by any of the manufacturer's facilities, it is added to the excess demand which will be outsourced during the period. Thus, we have a total of 15 decision variables (3 facilities and 5 products). Each simulation period is a week and the simulation runs over 1 year (i.e., 52 periods). We consider different levels of demand uncertainty, by varying the coefficient of variation χ across four scenarios: $\chi = 0.1, 0.3, 0.6$ and 0.9 ($\chi = 0.3$ coincides with the coefficient of variation that results from historical data; the other values were chosen to test the algorithms under sufficiently diverse conditions). All scenarios assume normal demand distributions for all product types; if a negative value is drawn in the simulations, the absolute value is used.

6.2.2. The (s,S) inventory control problem

Inventory management problems occur frequently in real business environments. One of the most studied inventory management policies in the simulation optimization literature is the (s, S) inventory policy (also known as *base stock policy*), which raises the inventory position (which equals the physical inventory plus outstanding orders minus backorders) to the order up to level S, as soon as it drops to or below the reorder point s. The (s, S) inventory setting we consider has been extensively studied in the literature (see e.g., Bashyam and Fu 1998, Kleijnen and Wan 2007). Recently, Angun and Kleijnen (2023) study this problem in a single-objective setting where the average total cost (consisting of the sum of inventory holding costs, and variable and fixed ordering costs) is to be minimized subject to a service level constraint (see Section 6.4 in their article).

We choose the input parameters and decision variables for the problem equal to the ones in Angun and Kleijnen (2023) and Kleijnen and Wan (2007). We consider 2 decision variables s and Q, where $S = s + Q$. The inventory position is reviewed periodically: the per-period demand D follows an exponential distribution with mean $\mu_D = 100$, while the replenishment lead time L follows an integer-valued Poisson distribution with mean $\mu_L = 6$ periods. The variable ordering cost $u = 1$, the holding cost $h = 1$, and the fixed ordering cost $K = 36$. The value of s needs to fall in the interval $[\mu_D \times \mu_L, 3 \times \mu_D \times \mu_L] = [600, 1800]$, while Q needs to fall in the interval $[42.5, 255]$. Due to the stochastic lead times, order crossovers can occur (i.e., orders are not necessarily received in the same order as they were placed), such that the problem is analytically intractable and stochastic simulation is needed to find the s and Q.

Table 3
Summary of the experimental scenarios for the demand allocation problem.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Kernel used in SK metamodels	Matérn: Eq. (8)			
Number of points in S ($\|S\|$)	125	96	93	99
Size of the true Pareto set $\|PS_t\|$	33	32	22	25
Coefficient of variation of demand	$\chi = 0.1$	$\chi = 0.3$	$\chi = 0.6$	$\chi = 0.9$

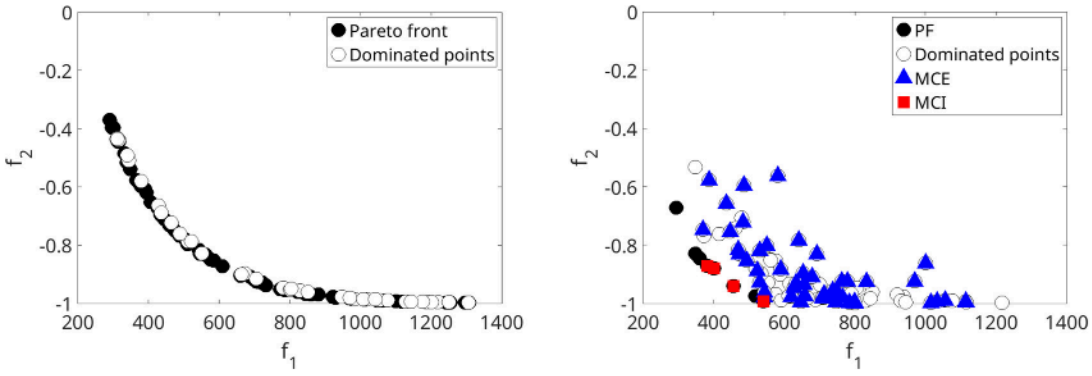


Fig. 5. Pareto front for the simultaneous minimization of cost (denoted f_1) and negative service level (denoted f_2) for the (s,S) inventory problem. The left panel shows the approximation of the true Pareto front, obtained from a simulation run with $P = 30000$ periods. The right panel illustrates the misclassification errors resulting at an arbitrary iteration of the same instance, when $P = 208$ periods.

Table 4
Summary of the experimental scenarios for the (s,S) inventory problem.

	Scenario 1	Scenario 2
Kernel used in SK metamodels	Matérn: Eq. (8)	
Number of points in S ($\|S\|$)	101	120
Size of the true Pareto set $\|PS_t\|$	62	36
Number of periods	208	832

We formulate the problem as bi-objective, where the goal is to minimize the expected total cost while simultaneously minimizing the negative service level. We consider 2 scenarios (see Table 4). The number of periods P differs across the scenarios: evidently, scenario 1 yields more noisy observations for the objectives than scenario 2, as it has fewer replications. Yet, the observed objective outcomes in scenario 2 are naturally *closer* to the true front, making the ranking and selection problem in this scenario equally challenging.

The trade-offs between the objectives are illustrated in Fig. 5. The left panel shows the approximation of the true Pareto front, obtained from a simulation run with P = 30 000. As originally recommended by Bashyam and Fu (1998), using this number of periods achieves a steady-state, ensuring that the remaining noise in the resulting objectives is negligible. This approximated Pareto front is the reference for evaluating the misclassification errors obtained by the different algorithms.

7. Results

In Section 7.1.1, we compare the performance of the SK-MORS algorithm against EQUAL allocation and the MOCBA algorithm (Chen & Lee, 2010) on the analytical test functions, where we also show that resampling points by simultaneously maximizing EHVD and PD outperforms a resampling approach based on the maximization of only one of those in isolation. In Section 7.1.2, we analyze the benefit of using the SK information instead of the sample means in the EQUAL allocation and MOCBA algorithms. We also provide a separate evaluation of the performance of the screening procedures in the supplementary material (see Appendix A). Finally, in Section 7.2 we show the performance on the two practical problems.

For each problem instance (analytical and practical), we run the algorithms 30 times on the same instance (i.e., 30 macroreplications); in the figures, we report the average APS value obtained at each iteration of the algorithms, along with the 95% confidence intervals. We set $B = |S|$ (i.e., the budget per iteration equals the cardinality of the set of points), to facilitate the comparison of our algorithm with standard EQUAL allocation, which uniformly distributes the replication budget among *all* alternatives (thus a budget of $B = |S|$ will run 1 additional replication per point per iteration). Furthermore, a value of $b_0 = 5$ is used to approximate the performance in all algorithms, and a value of $b_{max} = 100$ is used for MOCBA and SK-MORS. Table 5 summarizes the algorithms evaluated.

7.1. Performance on the analytical test functions

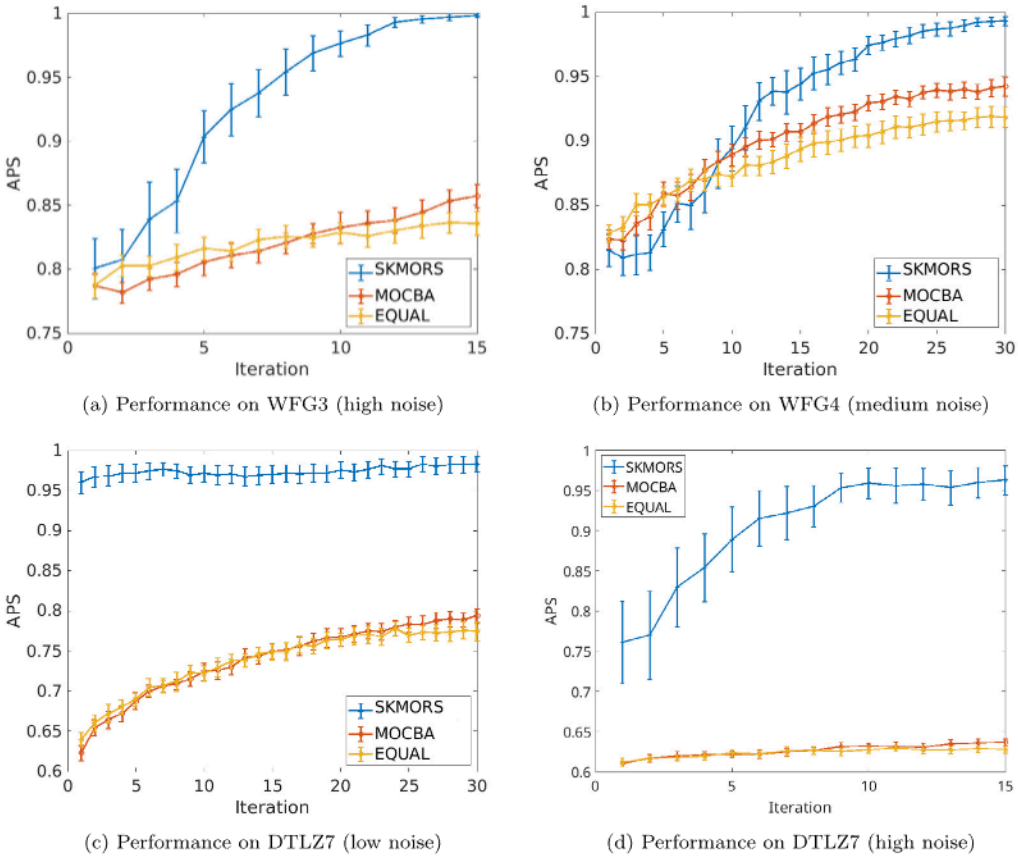
7.1.1. Proposed allocation criteria

Fig. 6 clearly shows that for all test problems considered, the proposed algorithm converges faster towards the correct identification of the true Pareto front. Moreover, the results in the supplementary material (see Table 7) show that when the budget is depleted, the proposed algorithm consistently yields lower number of true errors on average than its competitors. Note also in Fig. 6 that the superiority of MOCBA over EQUAL is not entirely clear for most scenarios, as in these MOCBA and EQUAL require a *much* larger budget to converge. On the other hand, SK-MORS shows consistent performance. In scenario DTLZ7 (Fig. 6(c) and (d)), the proposed approach realizes a remarkable jump in APS already at the beginning of the SK-MORS allocation; for scenario DTLZ7 with noise, the variability is significantly higher, but remains clearly superior to the competitors.

Fig. 7 shows that the best results eventually are achieved when both criteria are combined, as opposed to using the PD and EHVD separately (see Fig. 2). This is expected, given that they focus on different contributions (allocating extra replications to the most uncertain points, versus allocating replications to the points that are expected to change the estimated front). An advantage of the PD criterion is its very low computational cost, although the EHVD is shown to be more efficient in convergence.

Table 5
Overview of the different algorithms tested.

Algorithm	Description
SK-MORS	Algorithm 1.
SKMORS-PD	SK-MORS algorithm using only the PD criterion.
SKMORS-HV	SK-MORS algorithm using only the EHVD criterion.
EQUAL	Allocate replications uniformly to all sampled points.
EQUAL-SKi	EQUAL allocation using the SK predictions for identification.
MOCBA	Multi-objective Optimal Computing Budget Allocation algorithm (Chen & Lee, 2010).
MOCBA-SK	MOCBA algorithm using the SK predictions for allocation and identification.
MOCBA-SKi	MOCBA algorithm using the SK predictions only for identification.



(a) Performance on WFG3 (high noise)

(b) Performance on WFG4 (medium noise)

(c) Performance on DTLZ7 (low noise)

(d) Performance on DTLZ7 (high noise)

Fig. 6. Convergence to the true Pareto set: the proposed algorithm versus standard EQUAL allocation, and the state-of-the-art MOCBA procedure, for the 4 artificial test problems considered.

7.1.2. Impact of stochastic kriging

As evident from Fig. 8, feeding SK information to the identification phase of EQUAL and MOCBA (i.e., using SK predictions only when computing the misclassification errors), leads to a drastic improvement in both algorithms (referred to as EQUAL-SKi and MOCBA-SKi respectively); yet, they still show worse performance than the proposed method. The bad performance of MOCBA-SK (i.e., feeding SK information to the MOCBA algorithm for allocation and identification) is not entirely surprising, as the sample variance and prediction MSE do not reflect the same information (i.e., sampling variability and prediction error, respectively). Overall, it is clear that the efficient identification of the solutions with the true best expected performance is facilitated by exploiting the stochastic kriging information (as opposed to relying on the sample means). However, as shown in Table 6, fitting the metamodels incurs higher running times than EQUAL or MOCBA. In turn, the running time of these two worsens when they are assisted by metamodels. In practice, this trade-off is relative to the underlying simulation used: if the cost of evaluating the simulation is 10^2 or 10^3 times more expensive than fitting the metamodels, then the differences shown in Table 6 become irrelevant. In addition, using the proposed approach is beneficial when the replication budget is extremely limited, as it is shown to have a faster convergence rates (see e.g., Fig. 6).

7.2. Performance on the practical problems

Fig. 9(a)–(d) shows the results for the 4 scenarios considered for Practical Problem #1, and Fig. 9(e)–(f) shows the results for the 2 scenarios considered for Practical Problem #2. For Problem #1, it is evident that with increasing noise levels the method becomes more useful. For Problem #2, it is remarkable how the proposed algorithm is much closer to the maximum APS already from a very early stage of the allocation; this is evidently due to the use of the predictive models to determine the Pareto set. Moreover, in contrast with the results on the artificial test functions, for both problems MOCBA shows clear superiority to EQUAL (see e.g., Fig. 9(d) and (e)).

A relevant observation that is not visible from the figure is that all algorithms struggle to converge to zero errors, even with very large budgets of 100 or more iterations, because of several solutions with only marginally different performance (conversely, see the results in Fig. 6). Indeed, in practice we often encounter solutions with

Table 6
Average (std. err.) of the running time per iteration, in seconds, for the WFG4 problem.

SK-MORS	SK-MORS-PD	SK-MORS-HV	MOCBA	MOCBA-SKi	MOCBA-SK	EQUAL	EQUAL-SKi
60.1 (9.78)	21.6 (2.21)	58.6 (19.44)	0.5 (0.00)	18.01 (4.02)	19.25 (3.43)	0.00 (0.00)	25.2 (2.13)

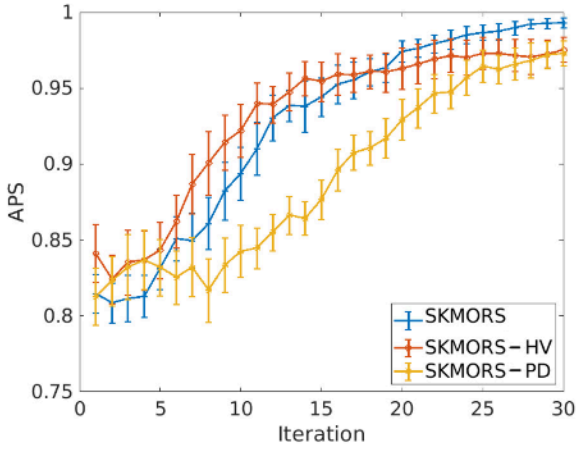


Fig. 7. Individual performance of the proposed resampling criteria on WFG4 (medium noise).

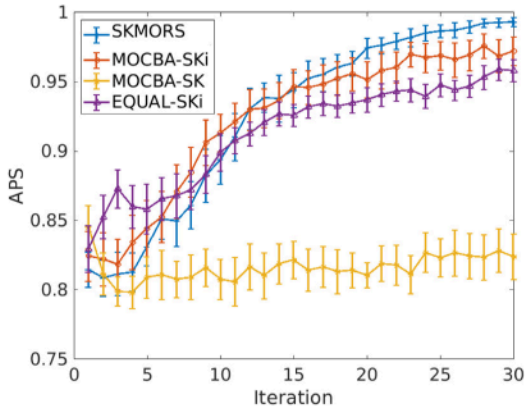


Fig. 8. Performance on WFG4 (medium noise).

marginally different performance, and thus also incorporating a *multi-objective indifference zone* procedure can be beneficial. Yet again, these remain very scarce in the literature.

8. Conclusions and future research

This paper proposed a bi-objective ranking and selection procedure designed to allocate additional replications to specific alternatives within a finite set, aiming to correctly identify the true non-dominated solutions. The procedure relies on two sampling criteria that make use of stochastic kriging metamodels. The first criterion, EHVD, focuses on solutions where additional replications are expected to produce a significant change in the estimated Pareto front. The second criterion, PD, focuses on improving the precision of the estimated performance, but without considering the dominance relationships between alternatives. By selecting points that simultaneously maximize both criteria, we obtain an algorithm that efficiently allocates the available replication budget in view of reducing the misclassification errors.

Fitting the metamodels implies more computational effort than EQUAL and MOCBA, especially in settings with high input dimensions. The exact computation of the metamodels beyond 15 input dimensions would require implementing an approximation (see e.g., Lu et al. 2020), or a dimensionality reduction technique (see e.g., Binois and Wycoff 2022), or using a different metamodel. While in practical settings this running time difference may not be significant, to reduce the computational cost we propose two screening procedures. The limited experiments show that the use of these procedures did not significantly impact the performance of the algorithm in terms of convergence. The choice of which one to use in practice will depend on the specific problem at hand. For example, if detecting small differences in performance for the observed non-dominated solutions is not crucial for the decision-maker, using Screening Band can be helpful to reduce the computational overhead significantly, while still considering well-performing solutions. Moreover, both procedures may be combined, first using Screening Box and then switching to Screening Band in later iterations.

In settings with high noise, the sample variance can be inaccurate with respect to the true variance if only a small number of replications is taken. Yet, the results obtained illustrate that the proposed algorithm consistently performs well, even if the (endogenous) noise on the objective values is increased by manipulating the uncertainty parameters. A key further work direction is to investigate the effect of using common random numbers (CRN) in our setting; while CRN are shown to inflate the MSE in Ankenman et al. (2010), in Chen et al. (2012) they are shown to be useful for simulation–optimization, and thus it may be beneficial in the MORS setting.

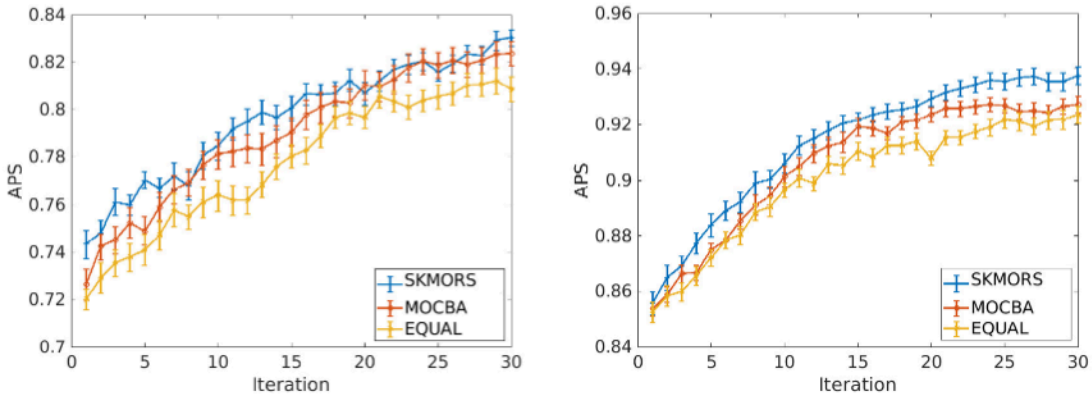
Finally, extending the algorithm to handle more objectives is conceptually straightforward, as metamodels can be constructed for any required number of objectives. However, as highlighted by Afsar et al. (2023), increasing the number of objectives leads to a dramatic rise in the number of misclassification errors. In fact, MORS procedures tend to allocate an excessively large replication budget to distinguish solutions that differ only marginally, which exacerbates when the number of alternatives reaches the hundreds or thousands, as is common in high-dimensional objective spaces. In this context, exploring parallelization of the resampling criteria or implementing batch resampling — such as identifying batches of solutions that collectively maximize the EHVD — could be highly beneficial. Moreover, research on scalability in objective space for problems involving many stochastic objectives remains in its infancy (see e.g., Cooper et al. 2020, Hunter et al. 2019). A core challenge is that the number of solutions needed to approximate the Pareto front grows exponentially with the number of objectives, and this, in turn, leads to a steep increase in the replication budget required.

CRediT authorship contribution statement

Sebastian Rojas Gonzalez: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Funding acquisition, Conceptualization. **Juergen Branke:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Inneke Van Nieuwenhuyse:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

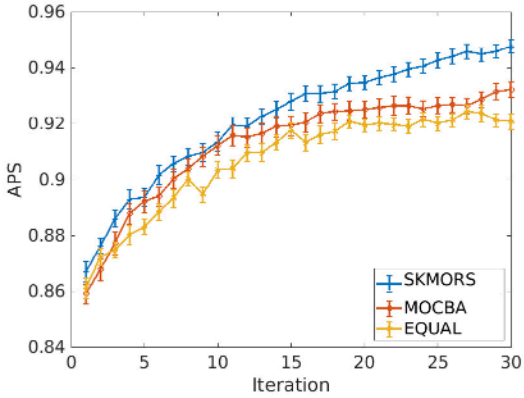
Acknowledgments

This research was supported by the Research Foundation-Flanders, grant number 12AZF24N. We are also thankful to Juan Ungredda from our industrial partner for providing the simulator from the chemical industry, and to Prof. Dr. Jack Kleijnen (Tilburg University) and Prof. Dr. Ebru Angun (Galatasaray University) for providing the latest manuscript version of Angun and Kleijnen (2023) and the base code for the (s, S) inventory simulation problem.

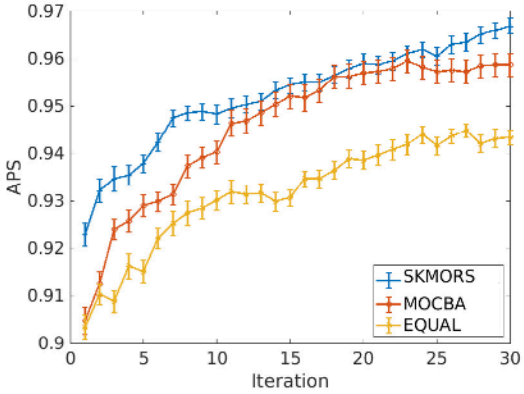


(a) Performance on Scenario 1, Problem #1

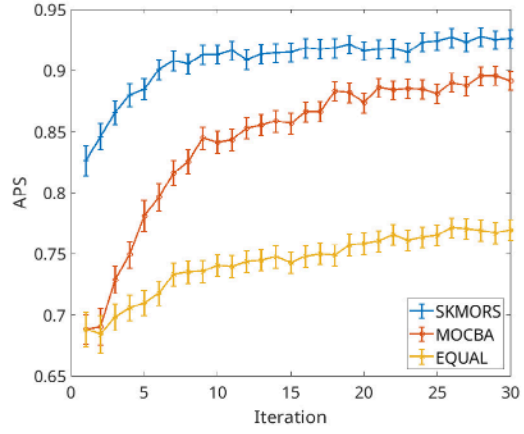
(b) Performance on Scenario 2, Problem #1



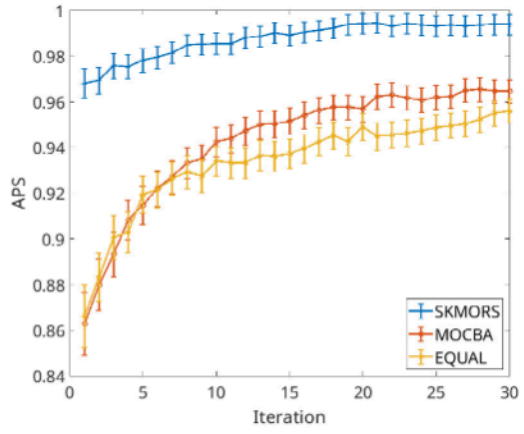
(c) Performance on Scenario 3, Problem #1



(d) Performance on Scenario 4, Problem #1



(e) Performance on Scenario 1, Problem #2



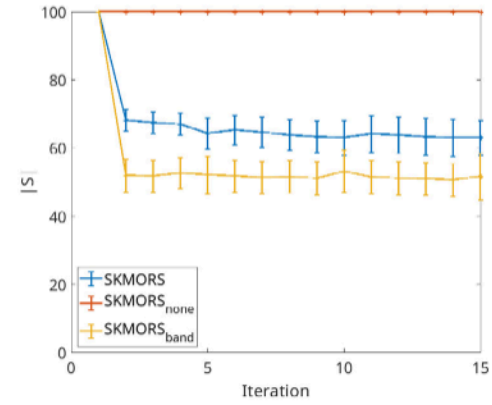
(f) Performance on Scenario 2, Problem #2

Fig. 9. Performance of SK-MORS vs. EQUAL and MOCBA for the 6 practical scenarios considered.

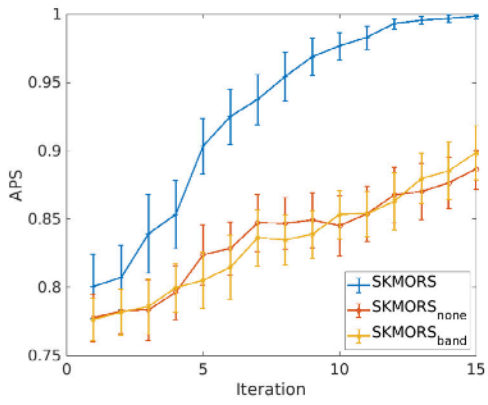
Appendix A. Screening band procedure

Here we propose an alternative screening procedure, named Screening Band, which encloses a confidence region using the upper confidence bounds of *all* the non-dominated points, not just the worst upper bound among all non-dominated points (as with Screening Box). The procedure is outlined in Algorithm 2, and the relative performance is compared in Fig. 10. SK-MORS$_{none}$ denotes an algorithm that does not screen out any points, and the metric $|S|$ shows the number of points that remain in the set after screening, at a given iteration.

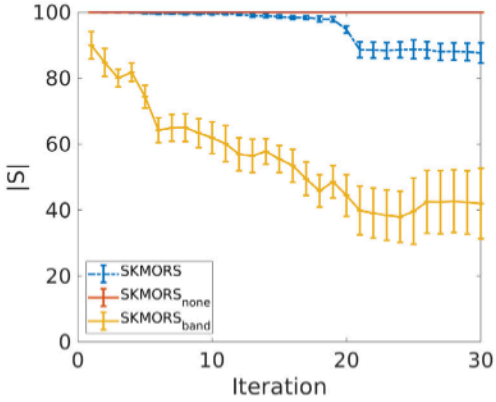
Both screening procedures succeed in sequentially reducing the number of candidate points considered as the observed performance becomes more accurate (see (a), (c) and (e) in Fig. 10); evidently, the biggest reduction is obtained with Screening Band. Such reduction significantly reduces the computational cost of executing the hypervolume calculations in Eq. (14) for every candidate point at each iteration, but at a higher risk of excluding true non-dominated points from resampling. Indeed, as is clear from Fig. 10(b), (d) and (f), the noise levels for the different scenarios have an impact on the performance of the proposed algorithm with respect to the convergence to the true front.



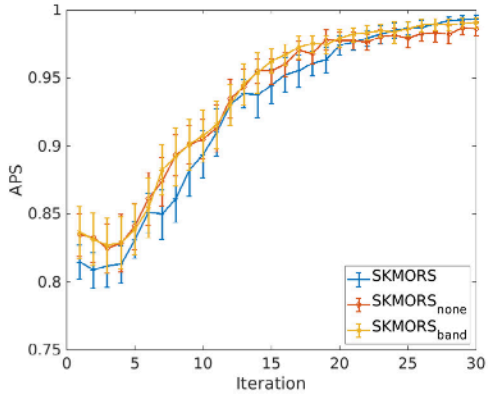
(a) Average number of points considered at each iteration for scenario WFG3 (high noise)



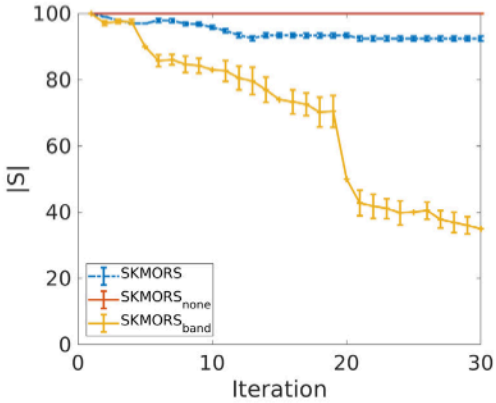
(b) Performance on WFG3 (high noise)



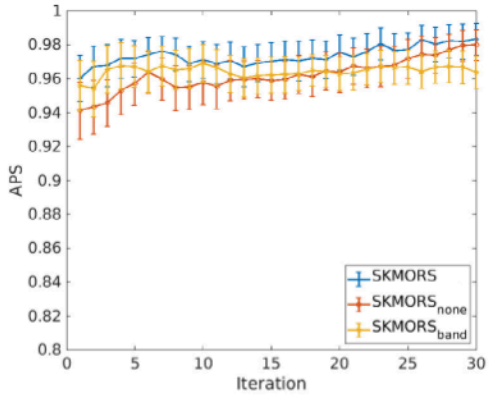
(c) Average number of points considered at each iteration for scenario WFG4 (medium noise)



(d) Performance on WFG4 (medium noise)



(e) Average number of points considered at each iteration for scenario DTLZ7 (low noise)



(f) Performance on DTLZ7 (low noise)

Fig. 10. Performance of the proposed screening procedures.

While for low and medium levels of noise no particular differences are observed between both proposed methods, for a high level of noise the performance of Screening Band clearly worsens. Moreover, in Fig. 10(e), we observe that the BAND procedure screens out a very large number of points, including some of the MCE errors. This happens because the noise is low and the dominated points are relatively far from the non-dominated points.

Appendix B. MOCBA allocation rules

The allocation rules of the simplified MOCBA algorithm (Chen & Lee, 2010) are summarized below. The following notation is used:

$\bar{f}_{ij}$: The sample mean of objective j in design i.

p_i : The design that dominates design i with the highest probability.

Algorithm 2 SCREENING BAND

Input: $\overline{UCB}$, $\overline{LCB}$, $\widehat{UCB}$ and $\widehat{LCB}$
$I = \emptyset$ ▷ Initialize set of clearly inferior points
$NP = S \setminus \overline{PS} \cap S \setminus \widehat{PS}$
procedure DISTANCE($\mathbf{f}_i, \mathbf{f}_k$)
  $h_{ik} = \sqrt{\sum_{j=1}^{m} [f_{ij} - f_{kj}]^2}$
  return h_{ik}
end procedure
for $i \in NP$ **do**
  $\bar{h}_{ik}$ =DISTANCE($\overline{LCB}_i, \overline{UCB}_k$), $\forall k \in \overline{PF}$
  $\hat{h}_{ik}$ =DISTANCE($\widehat{LCB}_i, \widehat{UCB}_k$), $\forall k \in \widehat{PF}$
end for
for $i \in NP$ **do**
  $u = \text{argmin}_{k \in \overline{PF}} \bar{h}_{ik}$
  $v = \text{argmin}_{k \in \widehat{PF}} \hat{h}_{ik}$
  for $j = 1 : m$ **do**
    if $\overline{LCB}_{ij} > \overline{UCB}_{uj}$ **and** $\widehat{LCB}_{ij} > \widehat{UCB}_{vj}$ **then**
      $I \cup \{\mathbf{x}_i\}$
      break
    end if
  end for
end for
return I

$j^i_{p_i}$: The objective j of p_i that dominates the corresponding objective of design i with the lowest probability.
τ_{ij} : The sample standard deviation of design i for objective j.
α_i : The budget allocation for design i.
S: The entire set of sampled points.
S_A: The subset of designs in S labeled as being dominated.
S_B: The subset of designs in S labeled as being non-dominated.

For any given design $g, h \in S_A$ and $d \in S_B$:

$$\alpha_h = \left(\frac{\tau_{hj^h_{p_h}} / \delta_{hp_hj^h_{p_h}}}{\tau_{gj^g_{p_g}} / \delta_{gp_gj^g_{p_g}}} \right)^2 \tag{16}$$

$$\alpha_d = \sqrt{\sum_{h \in D_d} \frac{\tau^2_{dj^h_d}}{\tau^2_{hj^h_d}} \alpha^2_h} \tag{17}$$

where

$$\delta_{ipj} = \bar{f}_{pj} - \bar{f}_{ij}$$

$$j^i_p = \underset{j \in \{1,\ldots,m\}}{\text{argmin}} \; P(\bar{X}_{pj} \le \bar{X}_{ij}) = \underset{j \in \{1,\ldots,m\}}{\text{argmax}} \frac{\delta_{ipj}|\delta_{ipj}|}{\tau^2_{ij} + \tau^2_{pj}}$$

$$p_i = \underset{\substack{p \in S \\ p \neq i}}{\text{argmax}} \prod_{j=1}^{m} P(\bar{f}_{pj} \le \bar{f}_{ij}) = \underset{\substack{p \in S \\ p \neq i}}{\text{argmin}} \frac{\delta_{ipj^i_p}|\delta_i pj^i_p|}{\tau^2_{ij^i_p} + \tau^2_{pj^i_p}}$$

$$S_A = \left\{ h | h \in S, \frac{\delta^2_{hp_hj^h_{p_h}}}{\tau^2_{hj^h_{p_h}} + \tau^2_{p_hj^h_{p_h}}} < \min_{i \in D_h} \frac{\delta^2_{ihj^i_h}}{\tau^2_{ij^i_h} + \tau^2_{hj^i_h}} \right\} \tag{18}$$

$$S_B = S \setminus S_A \tag{19}$$

$$D_h = \{ i | i \in S, p_i = h \}$$

$$D_d = \{ h | h \in S_A, p_h = d \}$$

Appendix C. Formulation of the demand allocation problem

In general, the problem and corresponding notation are defined as follows:

P_{ijst}: amount of product i produced on processing unit j at facility s in time period t (decision variable).
S_{isct}: supply of product i from facility s to sales region c in time period t (decision variable).
C_{ijs}: unit production cost of product i on processing unit j at facility s.
t_{sc}: unit transportation cost from facility s to sales region c.
Γ_{ict}: amount of product i demanded by sales region c in time period t that must be outsourced due to insufficient supply at company's facilities.
ξ_{ic}: unit cost for outsourcing production of product i demanded in sales region c.
I_{ist}: inventory level of product i at the end of time period t at facility s.
h_{ist}: cost for holding a unit of product i in inventory at facility s for the duration of time period t.
R_{ijst}: effective production rate for product i on processing unit j at facility s in time period t.
RL_{ijst}: run length of product i on processing unit j at facility s in time period t.
H_{jst}: amount of time available for production on process j at facility s during time period t.
ω_{ict}: forecasted demand for product i in sales region c at time period t.
Z_{ict}: binary variable that takes value 1 if demand in sales region c in time period t is met by means by internal production, and 0 otherwise.
M_b: maximum amount of production that can be outsourced per product type, per period and per sales region.
M_s: minimum amount of production that can be outsourced per product type, per period and per sales region.
χ: coefficient of variation of the demand.
N_c: total number of sales regions.
N_t: total number of time periods.

The bi-objective optimization problem can then be formulated as follows:

$$\min \quad \sum_{i,j,s,t} C_{ijs} P_{ijst} + \sum_{i,s,c,t} t_{sc} S_{isct} + \sum_{i,c,t} \xi_{ic} \Gamma_{ict} + \sum_{i,s,t} h_{ist} I_{ist}$$

$$\max \quad \frac{1}{N_c N_t} \sum_{c,t} Z_{ict}$$

subject to:

$$P_{ijst}, S_{isct}, \Gamma_{ict}, I_{ist} \ge 0 \qquad \text{non-negativity constraint}$$

$$P_{ijst} = R_{ijs}(RL_{ijst})$$

$$P_{ijst} \le R_{ijs} H_{jst} \qquad \text{production capacity constraints}$$

$$I_{ist} = I_{is(t-1)} + \sum_j P_{ijst} - \sum_c S_{isct} \quad \forall i, s, t \qquad \text{inventory balance constraint}$$

$$M_s(1 - Z_{ict}) \le \Gamma_{ict} \le M_b(1 - Z_{ict}) \qquad \text{outsource amount constraints}$$

Appendix D. Results for the 10 artificial test problems considered

Table 7 shows the final APS (std. error) for the 10 artificial problems tested. Here *final* refers to the performance when the replication budget has been depleted (determined by the number of iterations). In addition to the WFG3, WFG4 and DTLZ7 bi-objective scenarios considered in Section 6, here we also include ZDT1 and DTLZ2, which are also well-known test problems in the literature (Huband et al., 2006). From these results, we extracted 4 scenarios to show the convergence of the algorithms in Section 7.

Table 7
Performance of EQUAL, MOCBA and SK-MORS on the 10 different artificial test problems considered.

Problem	Decision variables	Noise level	Size of PS_{true}	Size of S	Iterations	Algorithm	APS (std. err.)
ZDT1	5	Medium	25	100	15	EQUAL	0.65 (0.1115)
						MOCBA	0.74 (0.0165)
						SK-MORS	0.90 (0.1124)
ZDT1	10	High	25	100	30	EQUAL	0.75 (0.0781)
						MOCBA	0.77 (0.0129)
						SK-MORS	0.94 (0.1350)
DTLZ2	5	Low	20	100	15	EQUAL	0.68 (0.0277)
						MOCBA	0.71 (0.0129)
						SK-MORS	0.96 (0.0645)
DTLZ2	8	High	20	100	30	EQUAL	0.49 (0.1025)
						MOCBA	0.54 (0.1414)
						SK-MORS	0.88 (0.1231)
DTLZ7	2	Low	50	100	30	EQUAL	0.76 (0.0210)
						MOCBA	0.78 (0.0222)
						SK-MORS	0.97 (0.0231)
DTLZ7	10	High	75	184	15	EQUAL	0.74 (0.0250)
						MOCBA	0.77 (0.0325)
						SK-MORS	0.97 (0.0322)
WFG3	5	Medium	20	100	15	EQUAL	0.85 (0.0199)
						MOCBA	0.90 (0.0250)
						SK-MORS	0.99 (0.0010)
WFG3	5	High	20	100	15	EQUAL	0.83 (0.0228)
						MOCBA	0.86 (0.0244)
						SK-MORS	0.98 (0.0009)
WFG4	5	Low	20	100	30	EQUAL	0.90 (0.0258)
						MOCBA	0.92 (0.0411)
						SK-MORS	0.97 (0.0078)
WFG4	5	Medium	20	100	30	EQUAL	0.91 (0.0222)
						MOCBA	0.93 (0.0331)
						SK-MORS	0.98 (0.0088)

References

Afsar, B., Fieldsend, J. E., Guerreiro, A. P., Miettinen, K., Rojas Gonzalez, S., & Sato, H. (2023). Many-objective quality measures. In D. Brockhoff, M. Emmerich, B. Naujoks, & R. Purshouse (Eds.), *Many-criteria optimization and decision analysis: state-of-the-art, present challenges, and future perspectives* (pp. 113–148). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-031-25263-1_5.

Andradóttir, S., & Lee, J. S. (2021). Pareto set estimation with guaranteed probability of correct selection. *European Journal of Operational Research*, *292*(1), 286–298.

Angun, E., & Kleijnen, J. (2023). *Constrained optimization in random simulation: Efficient global optimization and Karush-Kuhn-Tucker conditions (Revision of CentER DP 2022-022)*: *CentER discussion paper, CentER discussion paper Nr. 2023-010*.

Ankenman, B., Nelson, B. L., & Staum, J. (2010). Stochastic kriging for simulation metamodeling. *Operations Research*, *58*(2), 371–382.

Applegate, E. A., Feldman, G., Hunter, S. R., & Pasupathy, R. (2020). Multi-objective ranking and selection: Optimal sampling laws and tractable approximations via SCORE. *Journal of Simulation*, *14*(1), 21–40.

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

Bashyam, S., & Fu, M. C. (1998). Optimization of (s, S) inventory systems with random lead times and a service level constraint. *Management Science*, *44*, 243–256.

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.

Binois, M., Ginsbourger, D., & Roustant, O. (2015). Quantifying uncertainty on Pareto fronts with Gaussian process conditional simulations. *European Journal of Operational Research*, *243*(2), 386–394.

Binois, M., & Wycoff, N. (2022). A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, *2*(2), 1–26.

Boesel, J., Nelson, B. L., & Kim, S.-H. (2003). Using ranking and selection to "Clean Up" after simulation optimization. *Operations Research*, *51*(5), 814–825.

Branke, J., & Zhang, W. (2019). Identifying efficient solutions via simulation: myopic multi-objective budget allocation for the bi-objective case. *OR Spectrum*, *41*(3), 831–865.

Branke, J., Zhang, W., & Tao, Y. (2016). Multiobjective ranking and selection based on hypervolume. In T. Roeder, P. Frazier, R. Szechtman, E. Zhou, T. Huschka, & S. Chick (Eds.), *Proceedings of the 2016 winter simulation conference* (pp. 859–870). Piscataway, New Jersey: IEEE.

Chen, X., Ankenman, B. E., & Nelson, B. L. (2012). The effects of common random numbers on stochastic kriging metamodels. *ACM Transactions on Modeling and Computer Simulation*, *22*(2), 7:1–7:20.

Chen, C.-H., & Lee, L. H. (2010). *Stochastic simulation optimization*: *vol. 1*, Singapore: World Scientific, http://dx.doi.org/10.1142/7437.

Chick, S. E., Branke, J., & Schmidt, C. (2010). Sequential sampling to myopically maximize the expected value of information. *INFORMS Journal on Computing*, *22*(1), 71–80. http://dx.doi.org/10.1287/ijoc.1090.0327.

Cooper, K., Hunter, S. R., & Nagaraj, K. (2020). Biobjective simulation optimization on integer lattices using the epsilon-constraint method in a retrospective approximation framework. *INFORMS Journal on Computing*, *32*(4), 1080–1100.

Cox, D. D., & John, S. (1992). A statistical method for global optimization. In *[proceedings] 1992 IEEE international conference on systems, man, and cybernetics* (pp. 1241–1246). IEEE.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Di Pietro, A., While, L., & Barone, L. (2004). Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions. *Vol. 2*, In *Proceedings of the 2004 congress on evolutionary computation* (pp. 1254–1261). IEEE.

Fan, W., Hong, L. J., & Nelson, B. L. (2016). Indifference-zone-free selection of the best. *Operations Research*, *64*(6), 1499–1514.

Feldman, G., & Hunter, S. R. (2018). SCORE allocations for Bi-objective ranking and selection. *ACM Transactions on Modeling and Computer Simulation*, *28*(1), 7:1–7:28.

Feldman, G., Hunter, S. R., & Pasupathy, R. (2015). Multiobjective simulation optimization on finite sets: Optimal allocation via scalarization. In Y. Yilmaz, W. Chan, I. Moon, T. Roeder, C. Macal, & M. Rosseti (Eds.), *Proceedings of the 2015 winter simulation conference* WSC, (pp. 3610–3621). Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc..

Feliot, P., Bect, J., & Vazquez, E. (2017). A Bayesian approach to constrained single-and multi-objective optimization. *Journal of Global Optimization*, *67*(1–2), 97–133.

Fieldsend, J. E., & Everson, R. M. (2005). Multi-objective optimisation in the presence of uncertainty. *Vol. 1*, In *2005 IEEE congress on evolutionary computation* (pp. 243–250).

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*(1), 103–117.

Frazier, P. I. (2014). A fully sequential elimination procedure for indifference-zone ranking and selection with tight bounds on probability of correct selection. *Operations Research*, *62*(4), 926–942.

Hernández-Lobato, D., Hernández-Lobato, J. M., Shah, A., & Adams, R. P. (2016). Predictive entropy search for multi-objective Bayesian optimization. In *Proceedings of the 33rd international conference on international conference on machine learning - volume 48* ICML '16, (pp. 1492–1501). JMLR.

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*(5), 477–506.

Hughes, E. J. (2001). Evolutionary multi-objective ranking with uncertainty and noise. In *International conference on evolutionary multi-criterion optimization* (pp. 329–343). Springer.

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 Transactions on Modeling and Computer Simulation*, *29*(1), 7:1–7:36. http://dx.doi.org/10.1145/3299872.

Jalali, H., Van Nieuwenhuyse, I., & Picheny, V. (2017). Comparison of Kriging-based algorithms for simulation optimization with heterogeneous noise. *European Journal of Operational Research*, *261*(1), 279–301.

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

Jung, J. Y., Blau, G., Pekny, J. F., Reklaitis, G. V., & Eversdyk, D. (2004). A simulation based optimization approach to supply chain management under demand uncertainty. *Computers & Chemical Engineering*, *28*(10), 2087–2106.

Kim, S.-H., & Nelson, B. L. (2001). A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, *11*(3), 251–273.

Kim, S.-H., & Nelson, B. L. (2006). Selecting the best system. *vol. 13*, In *Handbooks in operations research and management science* (pp. 501–534). North Holland: Elsevier.

Kleijnen, J. P. C. (2015). *Design and analysis of simulation experiments* (2nd ed.). NY: Springer.

Kleijnen, J. P., & Wan, J. (2007). Optimization of simulated systems: OptQuest and alternatives. *Simulation Modelling Practice and Theory*, *15*(3), 354–362.

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*(9), 656–674.

Lu, X., Rudi, A., Borgonovo, E., & Rosasco, L. (2020). Faster kriging: Facing high-dimensional simulators. *Operations Research*, *68*(1), 233–249.

Miettinen, K. (1999). *Nonlinear multiobjective optimization*: *vol. 12*, Springer Science & Business Media.

Picheny, V., Wagner, T., & Ginsbourger, D. (2013). A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, *48*(3), 607–626.

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning (Adaptive computation and machine learning)* (1st ed.). Cambridge, Massachusetts, USA: The MIT Press.

Rojas-Gonzalez, S., Branke, J., & Nieuwehuyse, I. V. (2019). Multiobjective ranking and selection with correlation and heteroscedastic noise. In *2019 winter simulation conference* WSC, (pp. 3392–3403).

Rojas-Gonzalez, S., Jalali, H., & Nieuwenhuyse, I. V. (2020). A multiobjective stochastic simulation optimization algorithm. *European Journal of Operational Research*, *284*(1), 212–226.

Rojas-Gonzalez, S., & Van Nieuwenhuyse, I. (2020). A survey on kriging-based infill algorithms for multiobjective simulation optimization. *Computers & Operations Research*, *116*, Article 104869.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, *25*.

Staum, J. (2009). Better simulation metamodeling: The why, what, and how of stochastic kriging. In *Proceedings of the 2009 winter simulation conference* WSC, (pp. 119–133). IEEE.

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*(3), 533–544.

Teich, J. (2001). Pareto-front exploration with uncertain objectives. In *International conference on evolutionary multi-criterion optimization* (pp. 314–328). Springer.

Teng, S., Lee, L. H., & Chew, E. P. (2010). Integration of indifference-zone with multi-objective computing budget allocation. *European Journal of Operational Research*, *203*(2), 419–429.

Trautmann, H., Mehnen, J., & Naujoks, B. (2009). Pareto-dominance in noisy environments. In *Proceedings of the eleventh conference on congress on evolutionary computation* (pp. 3119–3126). Piscataway, NJ, USA: IEEE Press.

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.

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.